

Dottorato di Ricerca in Fisica  
Ciclo XXX

Settore Concorsuale di afferenza: 02/A1

Settore Scientifico Disciplinare: FIS/01

**Software Development and  
Detector Characterization of the  
EUCLID Near-Infrared Spectro-Photometer**

Presentata da: **Federico Fornari**

Coordinatore Dottorato:  
**Prof.ssa Silvia Arcelli**

Supervisore:  
**Dott.ssa Laura Patrizii**

Co-supervisori:  
**Dott. Jean-Claude Clemens  
Dott. Gabriele Sirri**

**Esame finale anno 2018**



# Contents

<b>Introduction</b>	<b>IV</b>
<b>1 The Euclid mission</b>	<b>1</b>
1.1 Einstein and the static Universe . . . . .	1
1.2 Big Bang cosmological model with dark matter . . . . .	4
1.3 The discovery of the accelerated expansion of the Universe . . . . .	6
1.4 Cosmological Constant and Dark Energy . . . . .	7
1.5 $\Lambda$ CDM and alternative models . . . . .	11
1.6 Euclid science objectives . . . . .	17
1.6.1 Gravitational shear . . . . .	18
1.6.2 BAO and the Formation of Large Scale Structures . . . . .	19
<b>2 Mission architecture</b>	<b>22</b>
2.1 Service Module . . . . .	23
2.2 Payload Module . . . . .	27
2.2.1 The VIS instrument . . . . .	30
2.2.2 The NISP instrument . . . . .	31
2.2.3 NISP instrumental performances . . . . .	34
2.2.4 NISP observation sequence . . . . .	34
2.2.5 NISP electronics architecture . . . . .	35
<b>3 NISP software development</b>	<b>38</b>
3.1 The NI-ICU . . . . .	38
3.2 The NI-ICU ASW . . . . .	39
3.2.1 ICU-ASW Architectural design . . . . .	41
3.2.2 PUS services . . . . .	44
3.2.3 Operational flow . . . . .	44
3.2.4 Operating modes and mode transitions . . . . .	46
3.2.5 The test equipment for NI-ICU ASW interface . . . . .	48
3.2.6 The MIL-STD-1553 interface . . . . .	50
3.2.7 HW/SW configuration for MIL-STD-1553 interface simulators . . . . .	53
3.3 S/C simulator SW design . . . . .	54
3.3.1 S/C simulator communication test . . . . .	56
<b>4 Euclid infrared detectors</b>	<b>70</b>

4.1	Near-Infrared detection principles . . . . .	70
4.2	The pixel matrix of the IR detectors . . . . .	76
4.3	Intrinsic properties of the detectors . . . . .	78
4.3.1	Total noise . . . . .	79
4.3.2	Gain . . . . .	79
4.3.3	Non-linearity . . . . .	80
4.3.4	Inter-Pixel Capacitance (IPC) . . . . .	81
4.3.5	Full Well Capacity (FWC) . . . . .	81
4.3.6	Persistence . . . . .	81
4.3.7	kTC noise . . . . .	81
4.4	The H2RG detectors of the NISP instrument . . . . .	82
4.4.1	The detection system: Silicon Chip Sensor (SCS) . . . . .	83
4.4.2	The Sensor Chip Assembly (SCA) . . . . .	83
4.4.3	The readout electronics: Sensor Chip Electronics (SCE) . . . . .	84
4.5	Readout modes for H2RG detectors . . . . .	86
4.5.1	The correlated double sampling (CDS) . . . . .	87
4.5.2	Fowler-M acquisition . . . . .	88
4.5.3	The acquisition of the “Up The Ramp” type (UTR) . . . . .	90
4.5.4	The acquisition of Multiaccumulated type (MACC) . . . . .	90
4.5.5	The choice of the readout modes for NISP . . . . .	92
4.5.6	The NISP MACC readout modes . . . . .	92
4.6	Readout noise reduction . . . . .	93
4.6.1	The common mode correction . . . . .	94
4.7	Flux calculation . . . . .	95
4.7.1	Case without readout error . . . . .	96
4.7.2	Case with readout error . . . . .	97
4.7.3	The maximum likelihood correction . . . . .	100
4.7.4	Conclusions on flux calculation . . . . .	101
4.8	NISP detectors characterization . . . . .	101
4.9	Data processing on ground and characterization objectives . . . . .	103
4.10	Requirements for the characterization . . . . .	104
4.10.1	Detector noise correction . . . . .	104
4.10.2	Conversion gain . . . . .	106
4.10.3	Inter-pixel capacitance . . . . .	107
4.10.4	Non-linearity . . . . .	108
4.10.5	Persistence . . . . .	108
4.10.6	Summary of requirements for the characterization . . . . .	109
4.11	Characterization setup . . . . .	109
4.11.1	The environment . . . . .	110
4.11.2	The <i>Pégase</i> and <i>Andromède</i> cryostats . . . . .	111
4.11.3	The setup for characterization in cryostats . . . . .	112
4.12	Characterization data monitoring and analysis . . . . .	118
4.12.1	Slow Control interface . . . . .	119
4.12.2	The HDF5 acquisition file format . . . . .	121

<i>CONTENTS</i>	III
4.12.3 User interaction with DAS . . . . .	126
4.12.4 Data quality check (QC) . . . . .	129
4.12.5 Analysis report . . . . .	129
<b>5 H2RG dark current analysis</b>	<b>135</b>
5.1 Dataset description . . . . .	135
5.2 Pixel selections . . . . .	138
5.3 Analysis results . . . . .	144
<b>6 Conclusions</b>	<b>146</b>
<b>Appendices</b>	<b>154</b>

---

## Introduction

---

For many years cosmology has basically been a theoretical science, but recently a new epoch has begun: the availability of a consistent amount of observational data has paved the way for the transition to a so-called “precision cosmology” era: it is now possible to extract relevant information about interesting physics from large sets of astrophysical measurements.

With the Planck satellite [1], a considerable step forward has been made in the exploration of the first evolutionary phases of the Universe, through the precise analysis of the *Cosmic Microwave Background* (CMB), a widespread radiation field which brings us information about the state of the Universe at very high redshift ( $z \sim 1100$ ), when baryons and photons decoupled from each other. Nevertheless, precise information is required about the more recent history of the Universe, at redshift below 2, corresponding to 10 billions of years ago, when apparently an accelerated expansion phase took place and the first matter structures started to form and cluster. Based on current data, about 95% of the energy density of the Universe is provided by two unknown physical entities: *dark matter* ( $\sim 27\%$ ) and *dark energy* ( $\sim 68\%$ ). Concerning dark energy, the question is whether this component results from modification of General Relativity (the addition of a cosmological constant in Einstein’s gravitational field equations), or from a field which dynamically evolves with the expansion of the Universe. Dark Matter is of a nature completely unknown: satellite and earth based observations, as well as experiments at accelerators, e.g. at the LHC, are searching for still undiscovered particles as DM candidates. Investigations at low redshifts are deemed fundamental for the determination of the nature of dark matter and dark energy. The best way to investigate the low redshift Universe is through *galaxy surveys*. The measurement of three-dimensional *galaxy clustering* by spectroscopic redshift surveys for H $\alpha$  line emission combined with the *Baryon Acoustic Oscillation* (BAO) pattern, make us capable to study the spatial geometry of the large scale Universe. Besides, taking into account the redshift-space distortions, the measurement of galaxy clustering provides an evaluation of their velocity distribution and constrains the shape of the matter power spectrum.

The gravitational lensing effect due to the matter which lays between a distant galaxy and the observer is responsible of small distortions in galaxy intrinsic shape. This “weak” effect called “*weak lensing*” can be measured in order to characterize the geometry of the Universe and the dark matter distribution. Since weak lensing is a little percentage effect relatively to the intrinsic shape of the galaxies, a large number of observations is needed for statistical analyses. Galaxy clustering and weak lensing are two complementary probes when measured in the same

portion of the sky, since they are sensitive to both visible and dark matter.

The Euclid Mission, approved by the European Space Agency, is a space mission planned to perform an extensive cosmic survey over a 6 years period, from 2020 to 2026. The satellite will be equipped with two instruments: a visible imager (VIS) and a near-infrared spectrophotometer (NISP) to measure the shape and redshift of galaxies over a large fraction of the extragalactic sky.

The Euclid mission is managed by a consortium comprising 125 institutions from 15 different countries, involving more than 1200 researchers. Italian teams are involved on data reduction, VIS and NISP instrument operations, on-board electronics production and relative software integration, in collaboration with national space industries. The INFN groups have the responsibility of the Assembly, Integration and Validation of the NISP Warm Electronics. This implies the integration of on-board software on the NISP Instrument Control Unit (ICU) and Data Processing Unit (DPU), as well as the execution of integrated tests between ICU and DPU.

This thesis has been carried out in collaboration with the Bologna INFN team participating in Euclid. I have contributed to the design and implementation of a test environment aimed at developing and testing the communication between the Spacecraft and the NISP Instrument Control Unit. By this environment the development, test and validation of ICU Application Software functionalities are made possible. I also developed the software simulating the Euclid Spacecraft commanding and responding protocols.

As part of my PhD activity, I spent six months at the CPPM in Marseille collaborating with the local group in charge of the characterization of NISP infrared detectors. I took part in data acquisition shifts during detector calibration campaigns and studied the dependence on temperature of infrared detectors' dark current.

This thesis is organized as follows:

- in chapter 1 the scientific motivations for the Euclid mission are discussed. The theoretical models of modern cosmology are recalled;
- in chapter 2 the Euclid satellite is described, with a specific focus on the NISP instrument and its Warm Electronics;
- in chapter 3 the architecture of the ICU Application Software is reported. The hardware equipment used to develop the software simulating the counterparts of the ICU communication interface (Spacecraft and DPU) and the first communication test are described;
- in chapter 4 the operational principles of the Euclid H2RG infrared detectors, their properties and characterization procedure are discussed;
- in chapter 5 the study of H2RG dark current dependence on operation temperature is reported;
- in chapter 6 the accomplished activities and results are summarized. Finally, near future plannings for the Euclid mission are outlined.

Euclid is a mission of European Space Agency (ESA) planned to investigate on the cosmological questions discussed in the Introduction chapter, by performing a spectroscopic and photometric galaxy survey. Euclid was approved in 2012 as the next Medium Class mission of the ESA Cosmic Vision 2015-2025 [2]. Its primary goal is to provide constraints in order to discriminate among our current theories about the accelerated expansion of the Universe. It will exploit mainly two probes previously quoted: *galaxy clustering* and *weak lensing*. Several hypotheses have been considered in order to explain the accelerated expansion, such as the fact that, as far as we may apply Einstein's theory at large scales even if we cannot do it at quantum level, we may accept that the properties of the empty space at macroscopic and astronomical scales are scale invariant even if they are not at quantum level, thus implying a cosmic accelerated expansion of the Universe not linked with the presence of any unknown energy fields [3]. The scale invariance of the macroscopic empty space can also account for a modification of the virial theorem implying no need for dark matter [4]. Currently the most plausible theoretical option is that the accelerated expansion has to be attributed to a completely unknown term, which has been named *dark energy*. But how did this theory come out?

## 1.1 Einstein and the static Universe

The General Relativity tensorial Einstein's equations for the gravitational field are [5]

$$R_{ij} - \frac{1}{2}g_{ij}R = \frac{8\pi G}{c^4}T_{ij} \quad (1.1)$$

where  $R_{ij}$  is the Ricci tensor,  $g_{ij}$  is the metric tensor,  $R = g^{ij}R_{ij}$  is the Ricci scalar and  $G = 6.67 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$  is the gravitational constant. The equations show how space and time (at left-hand side) are determined by the energetic content of the Universe, represented by the stress-energy tensor  $T_{ij}$  on the right-hand side. At first approximation, the stress-energy

tensor is considered as that of a perfect fluid in equilibrium:

$$T_{ij} = -pg_{ij} + (p + \rho c^2)u_i u_j \quad (1.2)$$

where  $p$  is the thermodynamical pressure,  $\rho$  is the fluid density and  $u^i$  is the fluid four-velocity. Under the assumption of the *Cosmological Principle*, the simplest form of metric satisfying the homogeneity and isotropy constraints is the Fridman-Robertson-Walker metric [6]:

$$g_{\alpha\beta} dx^\alpha dx^\beta \equiv ds^2 = c^2 dt^2 - a^2 \left( \frac{dr^2}{1 - kr^2} + r^2 d\Omega^2 \right). \quad (1.3)$$

where  $d\Omega^2 = d\theta^2 + \sin^2 \theta d\phi^2$  ( $\theta$  and  $\phi$  are the spherical angular coordinates).  $a = a(t)$  represents the Universe *scale factor* (by which all the spatial part of the metric is normalized as a function of time),  $k$  parametrizes the Universe space-time curvature; it can assume only the values -1 for hyperbolic, 0 for euclidean flat and +1 for spherical geometry. Concerning the stress-energy tensor, one could take into account any kind of complicated physics related to viscosity effects or thermal gradients, but, in the end, homogeneity and isotropy principles would be broken. If (1.1) are solved using (1.2) and (1.3), then from the initial 16 equations only 2 equations are independent [7]. The first, named *time-time* equation, is:

$$\ddot{a} = -\frac{4}{3}\pi G \left( \rho + \frac{3p}{c^2} \right) a \quad (1.4)$$

the second, called *space-space* equation, is:

$$a\ddot{a} + 2\dot{a}^2 + 2kc^2 = 4\pi G \left( \rho - \frac{p}{c^2} \right) a^2 \quad (1.5)$$

and by replacing (1.4) in (1.5) one finds:

$$-4\pi G \left( \frac{\rho}{3} + \frac{p}{c^2} \right) a^2 + 2\dot{a}^2 + 2kc^2 = 4\pi G \left( \rho - \frac{p}{c^2} \right) a^2 \Rightarrow \dot{a}^2 + kc^2 = \frac{8}{3}\pi G \rho a^2. \quad (1.6)$$

As a matter of fact, (1.4) and (1.6) are not truly independent from each other because of *cosmological adiabatic condition*:

$$d(\rho a^3 c^2) = -p da^3 \quad (1.7)$$

since, as a closed system, the Universe cannot undergo a process of heat exchange with another system. Equation (1.4) describing how the second derivative with respect to time of the scale factor is related to  $p$  and  $\rho$ , gives information on how the accelerated expansion of the Universe may be affected by  $p$  and  $\rho$ . The sign of the acceleration depends on the term  $\left( \rho + \frac{3p}{c^2} \right)$ , which is always positive since  $p$  and  $\rho$  are positive quantities by definition. Therefore  $\ddot{a} \neq 0$  unless  $p$  and  $\rho$  are both null. The conclusion is that the Universe is not static, an idea not so easy to accept at the beginning of the XX century. Einstein himself tried to make a stationary solution be possible by introducing a new term in his equations [7]:

$$R_{ij} - \frac{1}{2}g_{ij}R - \Lambda g_{ij} = \frac{8\pi G}{c^4}T_{ij}. \quad (1.8)$$

Einstein put it on the left-hand side, but if it is moved to the right side, then actually its meaning changes from space-time term to energetic content term (to be summed with the stress-energy tensor). The  $\Lambda$  coefficient is defined as *cosmological constant* and the gravitational field equations are re-writable in such a way that the stationarity term  $\Lambda g_{ij}$  gets absorbed by the stress-energy tensor:

$$R_{ij} - \frac{1}{2}g_{ij}R = \frac{8\pi G}{c^4}\tilde{T}_{ij} \quad (1.9)$$

where

$$\tilde{T}_{ij} \equiv T_{ij} + \frac{\Lambda}{8\pi G}c^4 g_{ij}. \quad (1.10)$$

Equivalently, one can define an *effective pressure*  $\tilde{p}$  and an *effective density*  $\tilde{\rho}$  which take into account the contribution given by  $\Lambda$ :

$$\tilde{T}_{ij} = -\tilde{p}g_{ij} + (\tilde{p} + \tilde{\rho}c^2)u_i u_j. \quad (1.11)$$

The contribution of  $\Lambda$  to the pressure term can be expressed as

$$\tilde{p} = p + p_\Lambda \equiv p - \frac{\Lambda}{8\pi G}c^4 \quad (1.12)$$

while for the density one has

$$\tilde{\rho} = \rho + \rho_\Lambda = \rho + \frac{\Lambda}{8\pi G}c^2. \quad (1.13)$$

How to stop the Universe expansion? By putting  $\ddot{a} = 0$ : now with (1.9) one has  $\tilde{\rho} = -\frac{3\tilde{p}}{c^2}$ , that is allowed to happen in principle, since  $\tilde{p}$  could be negative (on the contrary,  $\tilde{\rho}$  can only be positive). The stationarity condition imposes that also  $\dot{a} = 0$  must hold, so that from (1.6) in which (1.2) is substituted by (1.11) one obtains

$$\tilde{\rho} = \frac{3kc^2}{8\pi Ga^2}. \quad (1.14)$$

The Universe model elaborated by Einstein was matter-dominated. Matter is one of the possible components of the Universe energy density and its contribution at fixed volume  $V$  is given by

$$p_m = Nk_B T = \frac{\rho_m k_B T}{m_p} = \frac{k_B T}{m_p c^2} \rho_m c^2 \quad (1.15)$$

where  $N = \frac{\rho_m}{m_p}$  indicates that matter can be mainly considered as made by hydrogen ( $\rho_m$  is matter density,  $m_p$  is proton mass). Since in general for ordinary matter particles the condition  $m_p c^2 \gg k_B T$  holds, then  $p_m \simeq 0$ . It follows that

$$\tilde{p} = -\frac{\Lambda c^4}{8\pi G} \Rightarrow \tilde{p} = -\frac{\tilde{\rho} c^2}{3} \Rightarrow \tilde{p} = -\frac{kc^4}{8\pi Ga^2}. \quad (1.16)$$

From (1.16)  $\Lambda = \frac{k}{a^2}$  and

$$\rho = \tilde{\rho} - \frac{\Lambda c^2}{8\pi G} = \frac{3kc^2}{8\pi Ga^2} - \frac{kc^2}{8\pi Ga^2} = \frac{kc^2}{4\pi Ga^2} \quad (1.17)$$

hence, as it should be by definition, it is found that  $\rho > 0$ . From this, it follows that  $k = 1$ , so that  $\Lambda = \frac{1}{a^2}$ : Einstein's Universe results spherical and static with a cosmological constant given by  $\Lambda_E = \frac{4\pi G\rho}{c^2}$ . From here, through dimensional analysis, one can easily see that the cosmological constant is 1 over a squared length. After the Universe expansion was confirmed, Einstein abandoned the idea of a cosmological constant. However, nowadays this parameter seems correlated to the dark energy content of the Universe.

## 1.2 Big Bang cosmological model with dark matter

When in 1926 Hubble proved the expansion of the Universe [8],  $\Lambda$  was cancelled from the gravitational field equations, implying a Universe that had had a beginning and is continuously expanding. This has been a major advancement in astronomy that has allowed to change our vision of man's place in space. Around the year 2000, several missions aiming to the observation of the sky allowed the emergence of a cosmological model linking observed phenomena to a theory capable of describing them. The reference model for the evolution of the Universe is illustrated in Figure 1.1. It is based on the current observations of the Universe and on the "Big Bang" hypothesis. This model can be decomposed into different periods beginning 13.7 billions of year ago (Big Bang), with the Universe being homogeneous and isotropic:

- *primordial inflation* beginning  $10^{-44}$  seconds after the Big Bang (*Planck time*): during this period of rapid expansion dark matter particles as well as the first baryons and leptons are formed;
- *primary nucleosynthesis* begins  $\approx 1$  second after the Big Bang; it takes three minutes for the first light nuclei to form (deuterium, helium, lithium and beryllium) in a dense and hot plasma ( $\approx 10^9$  K);
- the *opaque Universe* is the period covering up to 380000 years after the Big Bang; at this time the Universe is expanding but results far too dense to be observable: photons mean free path is too small, they are absorbed by surrounding ionized medium very quickly;
- *recombination* begins 380000 years after the Big Bang, when the Universe reaches a fairly low temperature (3000 K) so that the energy distribution of the photons emitted at this time partly sets below the energy of ionization of the light nuclei; free electrons are captured by the nuclei allowing the creation of the first atoms; at this moment the Universe becomes transparent to radiation and the first emitted photons (in the infrared domain) constitute the diffuse cosmic microwave background (CMB);
- the *dark ages* cover the period during which no astrophysical process can produce electromagnetic radiation, from 380000 to 150 million years after the Big Bang;

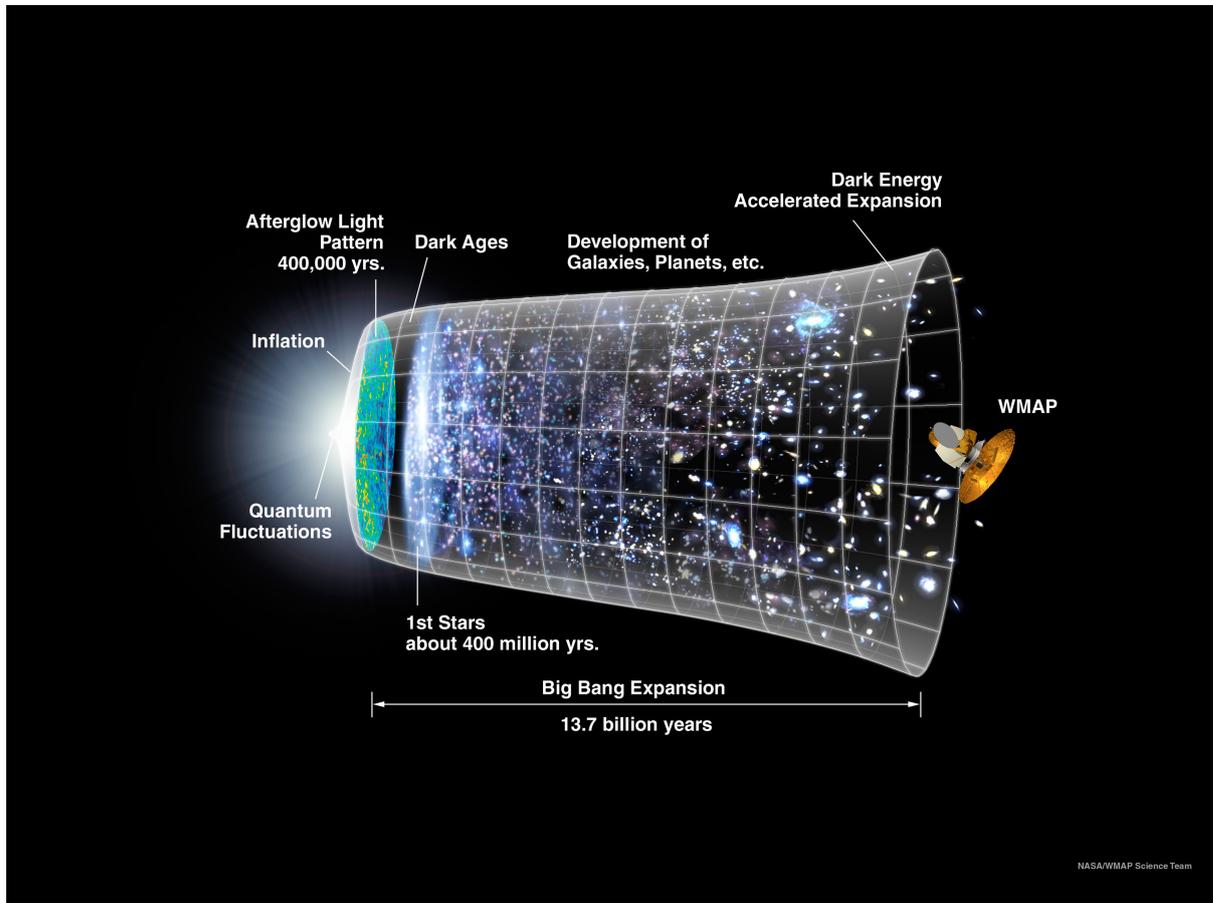


Figure 1.1: *Illustration of the evolution of the Universe from the Big Bang (Credits: NASA/WMAP Science Team).*

- the *formation of the structures* takes place about 1 billion years after the Big Bang: overdensities of matter (dark and ordinary) generate gravitational wells that attract matter in the form of gas clouds, leading to the formation of the first stars which then cluster in galaxies;
- the *current era* starts from 10 billion years after the Big Bang ( $\approx 2.7$  K) and is characterized by low density ( $\approx 10^{-29}$  g/cm<sup>3</sup>); photons from recombination are observable in the microwave domain ( $\lambda = 1.9$  mm).

It is in 1933 with the work of Fritz Zwicky that the first proof of a dark component of our Universe is provided [9]. Observing the dynamic mass of the Coma galaxy cluster, he noticed that it was superior to the expected mass inferred from the luminosities of the galaxies composing the cluster. This suggested a non-negligible mass emitting no light and forming part of the galaxy cluster. In 1970, measurements of the rotational speed of galaxies made by Vera Rubin

and Kent Ford supported this hypothesis by showing a radial velocity distribution of spiral galaxies much more flat than expected from the mass of luminous matter [10]. Another strong indication supporting the hypothesis of dark matter is the correlation between the measured CMB anisotropy and the distribution of large structures. Indeed, the observed CMB temperature fluctuation would generate gravitational potential wells unable to allow the formation of structures. By introducing dark matter, these gravitational potential wells become more consistent and, by simulating the distribution of the CMB, it is possible to see that they can reproduce a Universe matching today's observations of large structures of galaxies in filaments. In 1980, thanks to the main contributions of Peebles [11], Bond [12] and Blumenthal [13], the CDM (*Cold Dark Matter*) model emerged, based on a flat geometry for the large scale Universe with a hierarchical formation of structures influenced by a distribution of dark matter on smaller scales. This Universe model is consistent with the latest measurements of the Planck mission [1], which by measurements of the cosmic microwave background proved that the Universe is locally flat. The observation of clusters of galaxies implies a presence of dark matter, slowing down the expansion of the Universe by the action of gravitation.

### 1.3 The discovery of the accelerated expansion of the Universe

In 1999, the study of the remote supernovae (SN) of type Ia showed an inconsistency with respect to the CDM model [14]. SN Ia result from the accretion of the mass of a red giant by its companion at the stage of white dwarf. When the mass of the red giant accreted by the white dwarf reaches the mass limit of Chandrasekhar<sup>1</sup>, the star collapses and eventually a supernova is formed. The brightness of SN Ia, correlated to the mass of the star before the SN stage, is always the same, making it an excellent distance marker. The SN Ia are thus used as *standard candles*. In cosmology, the measurement of the distance of an object from the observer is related to the measurement of the Doppler shift in the spectrum of the object through the redshift  $z$ :

$$\lambda = \lambda_0(1 + z) \tag{1.18}$$

where  $\lambda_0$  is the wavelength at rest of the source and  $\lambda$  is the wavelength that is observed. From the curve of the brightness of SN Ia as a function of their redshift (and thus their distance), it turns out that the galaxies are moving away from us because of the expansion of the Universe [14]. However, they move faster than expected as shown in Figure 1.2, not consistently with the CDM model. The addition of a constant term  $\Lambda$  in the Einstein equation allows the model to be reconciled with the observations. This additional term into Einstein's equation results in the insertion of a negative pressure energy density called *dark energy*. The characteristic size of the angular fluctuations of the CMB intensity, of the order of 1 degree on the sky, is correlated with the spatial curvature of the Universe, which is found compatible with  $k = 0$  [16], showing that the Universe is geometrically locally flat. The addition of the cosmological constant bases the so-called "concordance model" or  $\Lambda$ CDM. In Figure 1.3 the results based on the observations of the cosmic microwave background, of the baryonic acoustic oscillations and of SN Ia are plotted

---

<sup>1</sup>The maximum mass that the electron degeneracy pressure of an object can withstand before gravity becomes strong enough to make the electrons collapse on protons.

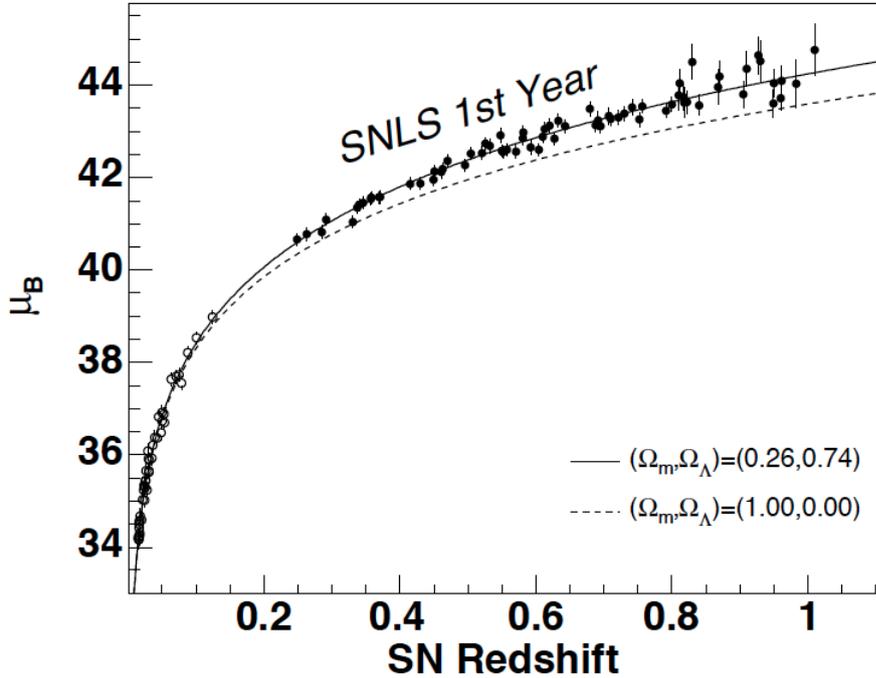


Figure 1.2: *Hubble diagram of SNLS (SN Legacy Survey) supernovae for 2 distributions with different values of matter density parameter  $\Omega_m$  and dark energy density parameter  $\Omega_\Lambda$  [15].*

together: the plot shows the current constraints on  $\Omega_m$  and  $\Omega_\Lambda$  cosmological parameters.

## 1.4 Cosmological Constant and Dark Energy

Let us now see how  $\Lambda$  was recovered to explain the presence of an accelerated expansion ( $\ddot{a} > 0$ ). Observations made in the early 1970s [18] showed concentrations of luminous sources at constant redshifts  $z$ , in particular of quasars at  $z = 2$ . A cosmological constant in the gravitational field equations was then in order to allow for the presence of consistent time intervals,  $\Delta t$ , when the Universe results stationary during its evolution (the so-called *Lemaître model*). The presence of  $\Lambda$  leads to an inflection point in the plot of the scale factor as a function of time (see Figure 1.4), so that a relative phase of stationarity in  $a(t)$  is followed by a parable branch with upward concavity. Anyway, the model was soon abandoned owing to the fact that the concentration of quasars at  $z = 2$  was found out to be only a bias due to selection effects.

In order to see that the presence of  $\Lambda$  is responsible for the acceleration, we consider for a while a Universe model in which there is no cosmological constant (the so-called *Fridman model*) and the gravitational field equations are (1.4) and (1.6), correlated through the adiabatic condition.

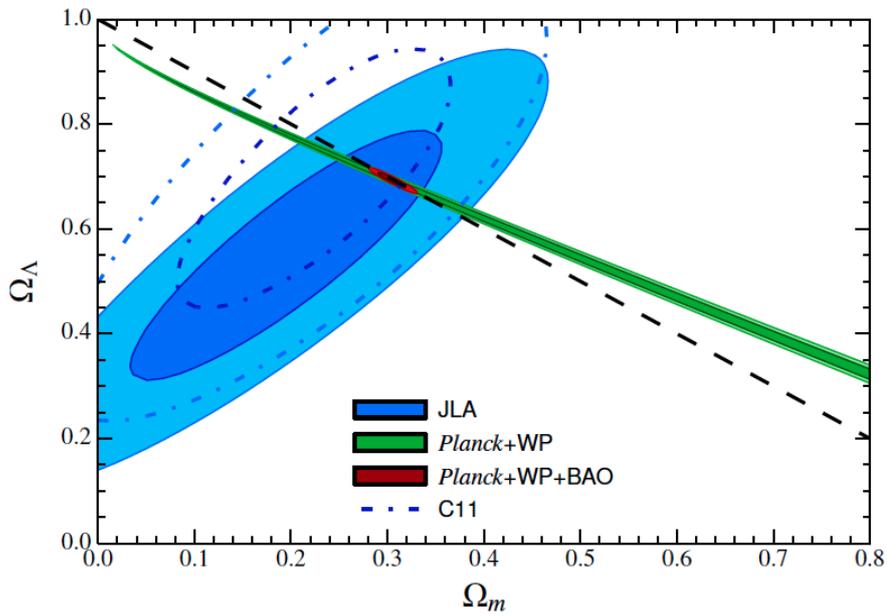


Figure 1.3: Confidence contours for cosmological parameters  $\Omega_m$  and  $\Omega_\Lambda$  obtained with observations from SNIa (blue), CMB (Planck+WMAP, green) and the combination with BAO measurements (red) [17].

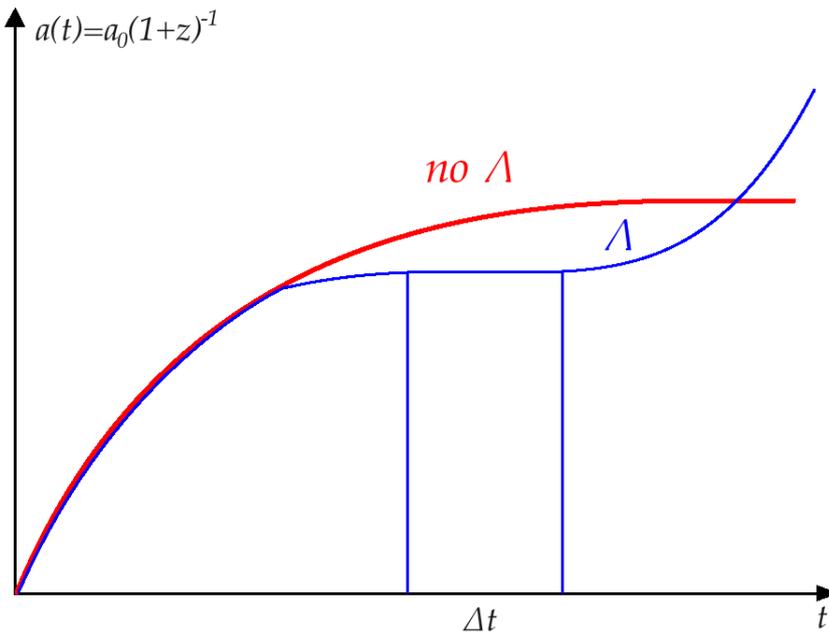


Figure 1.4: The introduction of a cosmological constant in the gravitational field equations (Lemaître model) makes the scale factor evolution undergo relevant stationarity phases.

Dividing (1.6) by  $a_0^2$  (where the “0” subscript stands for “present time”) one obtains

$$\frac{\dot{a}^2}{a_0^2} - \frac{8}{3}\pi G \rho \frac{a^2}{a_0^2} = -\frac{kc^2}{a_0^2}. \quad (1.19)$$

By introducing the *Hubble parameter*  $H \equiv \frac{\dot{a}}{a}$  ( $H_0 \equiv \frac{\dot{a}_0}{a_0}$ ) and the *critical density*  $\rho_{crit.} \equiv \frac{3H^2}{8\pi G}$  ( $\rho_{0\,crit.} \equiv \frac{3H_0^2}{8\pi G}$ ), (1.19) becomes

$$\frac{\dot{a}^2}{a_0^2} - \frac{\rho}{\rho_{0\,crit.}} H_0^2 \left(\frac{a}{a_0}\right)^2 = -\frac{kc^2}{a_0^2}.$$

At present time ( $a = a_0$ ,  $\dot{a} = \dot{a}_0$  and  $\rho = \rho_0$ ), one finds

$$H_0^2 - \frac{\rho_0}{\rho_{0\,crit.}} H_0^2 = -\frac{kc^2}{a_0^2} \Rightarrow H_0^2 \left(1 - \frac{\rho_0}{\rho_{0\,crit.}}\right) = -\frac{kc^2}{a_0^2}. \quad (1.20)$$

Equation (1.20) reveals the meaning of the  $\rho_{0\,crit.}$  parameter: if  $k = 0$ ,  $\rho_0 = \rho_{0\,crit.}$ , if  $k = +1$ ,  $\rho_0 > \rho_{0\,crit.}$ , and finally if  $k = -1$ ,  $\rho_0 < \rho_{0\,crit.}$ . This is a very important aspect: the global geometry of the Universe is directly linked to its energy density content. Conventionally, the density parameter  $\Omega \equiv \frac{\rho}{\rho_{crit.}}$  is introduced. As a result, one has

$$\begin{cases} \Omega_0 = \frac{\rho_0}{\rho_{0\,crit.}} = 1 \Rightarrow \text{Flat Universe} \\ \Omega_0 > 1 \Rightarrow \text{Spherical (closed) Universe} \\ \Omega_0 < 1 \Rightarrow \text{Hyperbolic (open) Universe} \end{cases}.$$

$\rho_{crit.}$  is a parameter which depends on  $H$ . From experimental observations  $\rho_{0\,crit.} = 1.9 \times 10^{-29} h^2 \text{ g/cm}^3$ , with  $h \simeq 0.7$  (uncertainty over  $H_0$  measures [19]). Let us introduce the cosmological constant in (1.4):

$$\begin{aligned} \ddot{a} &= -\frac{4}{3}\pi G \left[ \rho + \frac{\Lambda c^2}{8\pi G} + \frac{3p}{c^2} - \left(-\frac{3\Lambda c^4}{8\pi G c^2}\right) \right] a = -\frac{4}{3}\pi G \left(\rho + \frac{3p}{c^2}\right) a - \frac{4}{3}\pi G \left(\frac{\Lambda c^2}{2\pi G}\right) a \Rightarrow \\ \Rightarrow \ddot{a} &= -\frac{4}{3}\pi G \left(\rho + \frac{3p}{c^2}\right) a - \frac{2}{3}\Lambda c^2 a \end{aligned} \quad (1.21)$$

and in (1.6):

$$\dot{a}^2 + kc^2 = \frac{8}{3}\pi G a^2 \left(\rho + \frac{\Lambda c^2}{8\pi G}\right) \Rightarrow \dot{a}^2 + kc^2 = \frac{8}{3}\pi G a^2 \rho + \frac{\Lambda c^2 a^2}{3}. \quad (1.22)$$

By dividing (1.22) by  $a_0^2$

$$\frac{\dot{a}^2}{a_0^2} - \frac{8}{3}\pi G \frac{a^2}{a_0^2} \rho - \frac{\Lambda c^2}{3} \frac{a^2}{a_0^2} = -\frac{kc^2}{a_0^2}$$

at present time one finds

$$H_0^2 - H_0^2 \frac{\rho_0}{\rho_{0\,crit.}} - \frac{\Lambda c^2}{3} = -\frac{kc^2}{a_0^2}$$

and by defining  $\rho_\Lambda \equiv \frac{\Lambda c^2}{8\pi G}$  one has

$$H_0^2 \left( 1 - \frac{\rho_0}{\rho_{0 \text{ crit.}}} \right) - \frac{8\pi G \rho_\Lambda H_0^2}{3} = -\frac{kc^2}{a_0^2} \Rightarrow H_0^2 \left( 1 - \frac{\rho_0}{\rho_{0 \text{ crit.}}} - \frac{\rho_\Lambda}{\rho_{0 \text{ crit.}}} \right) = -\frac{kc^2}{a_0^2}. \quad (1.23)$$

As it can be seen, the result is an equation analogous to (1.20), so that by naming  $\Omega_{0\Lambda} \equiv \frac{\rho_\Lambda}{\rho_{0 \text{ crit.}}}$ , (1.23) becomes

$$H_0^2 [1 - (\Omega_0 + \Omega_{0\Lambda})] = -\frac{kc^2}{a_0^2} \quad (1.24)$$

and it can be observed that

$$\begin{cases} \Omega_0 + \Omega_{0\Lambda} = 1 \Rightarrow \text{flat curvature} \\ \Omega_0 + \Omega_{0\Lambda} > 1 \Rightarrow \text{spherical curvature} \\ \Omega_0 + \Omega_{0\Lambda} < 1 \Rightarrow \text{hyperbolic curvature} \end{cases} .$$

From current data [1] [16], we know that  $k$  is very close to 0, therefore  $\Omega_0 + \Omega_{0\Lambda} = 1$ . Also we know that the energy density contribution from baryonic and dark matter cannot exceed 30%, so that it is not possible for  $\Omega_0$  to assume a value greater than 0.3. This fact constrains the  $\Omega_{0\Lambda}$  term to be close to 0.7. What physical entity could be responsible for such a great share of the Universe energy density? At present nobody knows, and so the concept of *dark energy* has been introduced.

To conclude this discussion, let us show how the cosmological constant determines an acceleration in the evolution of the scale factor. Recalling the definitions of effective pressure and energy density in (1.12) and (1.13), and taking into account the fact that presently the dominant contributions to Universe energy density and pressure of the Universe are given by baryonic and dark matter ( $\Omega_{0,m} \simeq 0.3$ ), the *time-time* field equation becomes

$$\ddot{a} = -\frac{4}{3}\pi G \left( \tilde{\rho} + \frac{3\tilde{p}}{c^2} \right) a \quad (1.25)$$

where  $\tilde{\rho} = \rho_m + \rho_\Lambda = \rho_m + \frac{\Lambda c^2}{8\pi G}$  and  $\tilde{p} = p_m + p_\Lambda = p_m - \frac{\Lambda c^4}{8\pi G}$ . Therefore

$$\begin{aligned} \ddot{a} &= -\frac{4}{3}\pi G a \left( \rho_m + \frac{3p_m}{c^2} + \frac{\Lambda c^2}{8\pi G} - \frac{3\Lambda c^2}{8\pi G} \right) = -\frac{4}{3}\pi G a \left( \rho_m + 0 - \frac{\Lambda c^2}{4\pi G} \right) = \\ &= -\frac{4}{3}\pi G \rho_m a + \frac{\Lambda a c^2}{3} \end{aligned} \quad (1.26)$$

hence, by introducing the *deceleration parameter*  $q \equiv -\frac{\ddot{a}}{\dot{a}^2} a$  [7], one has

$$q_0 = -\frac{\ddot{a} a^2}{a \dot{a}^2} \Big|_{z=0} = -\frac{\ddot{a}}{a} \frac{1}{H_0^2} = \frac{1}{H_0^2} \left( \frac{4}{3}\pi G \rho_{0,m} - \frac{\Lambda c^2}{3} \right) = \frac{\Omega_{0,m}}{2} - \frac{\Lambda c^2}{3H_0^2} \equiv \frac{\Omega_{0,m}}{2} - \Omega_{0\Lambda}. \quad (1.27)$$

By inserting the data known from measures, one finds

$$q_0 \simeq \frac{0.3}{2} - 0.7 \simeq -0.6 \quad (1.28)$$

so that the deceleration turns out to be negative: it is an acceleration. That strictly depends on the presence of the  $\Omega_{0\Lambda}$  term, which represents the contribution of the cosmological constant.

## 1.5 $\Lambda$ CDM and alternative models

Let us imagine a cosmic fluid exclusively made of a cosmological constant component, so that  $\rho = p = 0$  and  $\tilde{p} = p_\Lambda$ ,  $\tilde{\rho} = \rho_\Lambda$ . In this case, recalling (1.12) and (1.13), one has

$$\rho = p = 0 \Rightarrow \tilde{p} = -\frac{\Lambda c^4}{8\pi G} = -\tilde{\rho}c^2 \Rightarrow p_\Lambda = -\rho_\Lambda c^2. \quad (1.29)$$

The general equation of state for a component of the Universe can be written as

$$p = w\rho c^2. \quad (1.30)$$

The proportionality parameter  $w$  between  $p$  and  $\rho$  may depend on redshift, so  $w = w(z)$ . One of the goals of Euclid is to measure  $w_\Lambda(z)$  with 2% accuracy on its constant part and 10% on its redshift-dependent part. The way to provide limits over  $w_\Lambda(z)$  is through an estimate of the gravitational lensing probability for distant objects.

The *comoving volume* included between redshift values  $z$  and  $z+dz$  (i.e. the volume determined by three-dimensional distances between astrophysical objects located at redshifts between  $z$  and  $z+dz$ , divided by three times the ratio of the scale factor then to now) increases with increasing  $\Omega_{0\Lambda}$ . This is because of (1.27): as  $\Omega_{0\Lambda}$  increases, the value of the acceleration becomes greater, so that at fixed time, light has to travel a longer distance to reach us from a specific source position (see Figure 1.5). One of the effects depending on the variation of the comoving volume (the number of galaxies in it being fixed), is the probability to observe gravitational lensing phenomena. Euclid will search for the presence of gravitational lenses, estimate the redshift

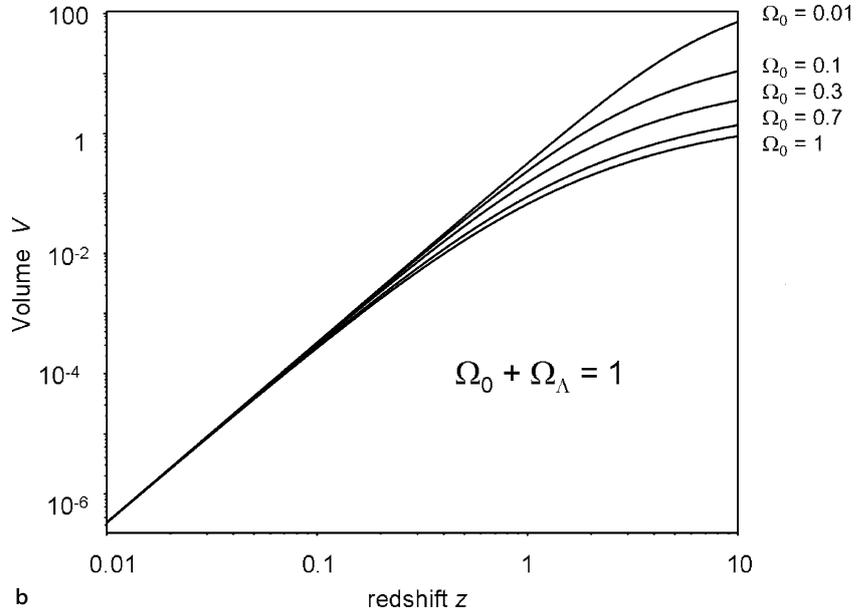


Figure 1.5: Variation of comoving volume with redshift in a Universe in which  $\Omega_0 + \Omega_{0\Lambda} = 1$ .

of the lensed object and calculate the probability for the lensing phenomenon to happen: in this way, limits over  $\Omega_{0\Lambda}$  and hence over  $w_\Lambda(z)$  can be derived. As pointed out with (1.29), the theoretical constant value for  $w_\Lambda(z)$  is -1. If this is the case, a cosmological constant is present in the gravitational field equations accounting for the dark energy density contribution together with *Cold Dark Matter*: as mentioned earlier, this so-called  $\Lambda$ CDM model is now regarded as the theoretical formulation which fits better old dark matter observational data, General Relativity and particle physics.

The compatibility of Cold Dark Matter (non-relativistic DM particles), rather than Hot Dark Matter (relativistic DM particles), with observational data derives from Jeans theory and dark matter power spectrum. Jeans mass  $M_J$  is the mass limit for a gravitational perturbation to collapse; for CDM,  $M_J^{cold} \sim 10^{5\div 6} M_\odot$ , while for HDM  $M_J^{hot} \sim 10^{15\div 16} M_\odot$ , an order of magnitude greater than average galaxy clusters [7]. Dark matter power spectrum (Figure 1.6) shows how CDM is able to justify the presence of smaller objects in the early Universe allowing for a *bottom-up* formation of larger structures, while HDM cannot account for the early formation of smaller scale structures.

It is also possible to observe how the neutrino mass hierarchy reflects on the process of cosmic structure formation. Depending on their mass, neutrinos have chances to travel longer

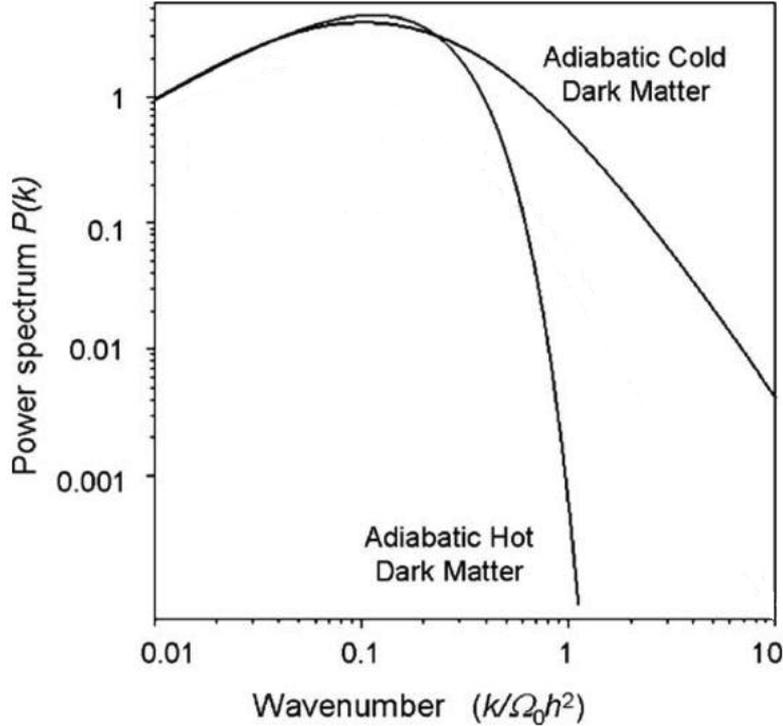


Figure 1.6: *Dark matter power spectrum  $P(k)$  for Hot and Cold adiabatic model. Normalization has been chosen to make the power spectra take the same value at large scales (small  $k$  values). The wave number  $k$  is measured in  $\text{Mpc}^{-1}$ .*

distances, hence affecting structure formation in the early Universe. A more or less intense suppression effect can be observed in the matter power spectrum over a wide range of scales and in particular at small scales, thus constraining the sum of neutrino masses. The matter power spectrum can help to discriminate between normal and inverted neutrino mass hierarchy (see Figure 1.7): the former prescribes the presence of two light and one heavy neutrinos, while the latter one light and two heavy neutrinos. In order to distinguish between the two hierarchies, the sum of neutrino masses has to be determined with a precision comparable with the maximum mass difference between neutrino types,  $\sim 0.06$  eV [2]. If the total mass of neutrinos  $m_\nu$  turns out to be smaller than 0.1 eV, the precision for expected Euclid measurement,  $\Delta m_\nu < 0.03$  eV, will allow to discriminate between the two mass orderings. The  $\Lambda$ CDM model does not solve all present issues, as the so-called *coincidence problem*. From the adiabatic condition (1.7):

$$\begin{aligned} d(\rho c^2 a^3) &= -p da^3 = -w\rho c^2 da^3 \Rightarrow d(\rho a^3) = -w\rho da^3 \Rightarrow d\rho a^3 + \rho da^3 = -w\rho da^3 \Rightarrow \\ \Rightarrow \rho(1+w) da^3 &= -d\rho a^3 \Rightarrow -\frac{d\rho}{\rho} = (1+w)\frac{da^3}{a^3} \Rightarrow -\ln \rho \propto (1+w)\ln a^3 \Rightarrow \rho \propto a^{-3(1+w)}. \end{aligned}$$

Recalling that at a given time  $1+z = \frac{a_0}{a}$  (with  $z$  the corresponding redshift), one finds out that  $\rho \propto (1+z)^{3(1+w)}$ . In case of matter dominance,  $w = 0$ , so  $\rho_m \propto a^{-3}$ . Since the radiation contribution to energy density is currently  $\Omega_{0,r} \sim 10^{-5}$ , it can be neglected. For the cosmological constant  $w = -1$ , so that  $\rho_\Lambda \propto const.$ , i.e.  $\rho_\Lambda = \rho_{0\Lambda}$ . The energy density for the cosmic fluid component associated to  $\Lambda$  is constant as a function of time. Summarizing, one has

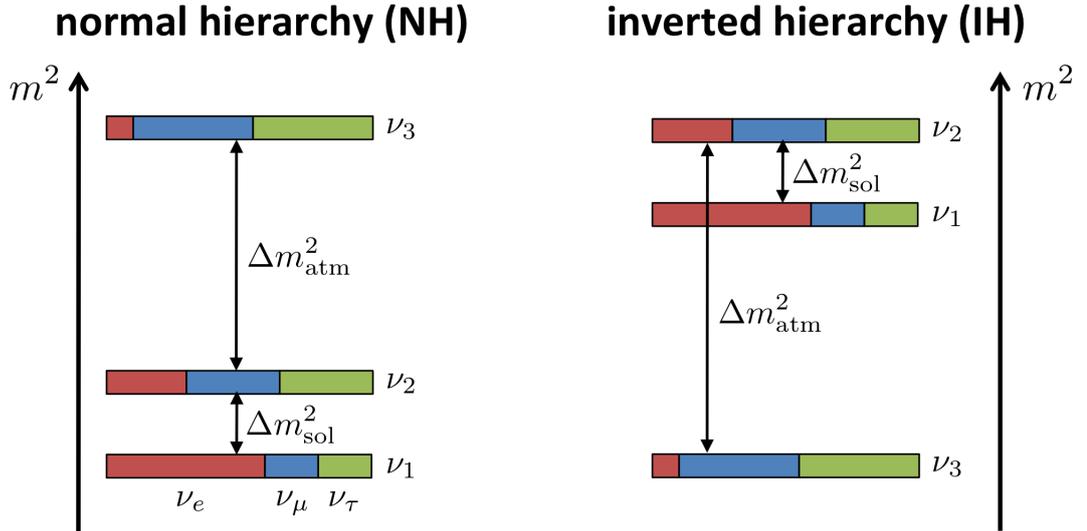


Figure 1.7: Normal and inverted hierarchy of neutrino masses.

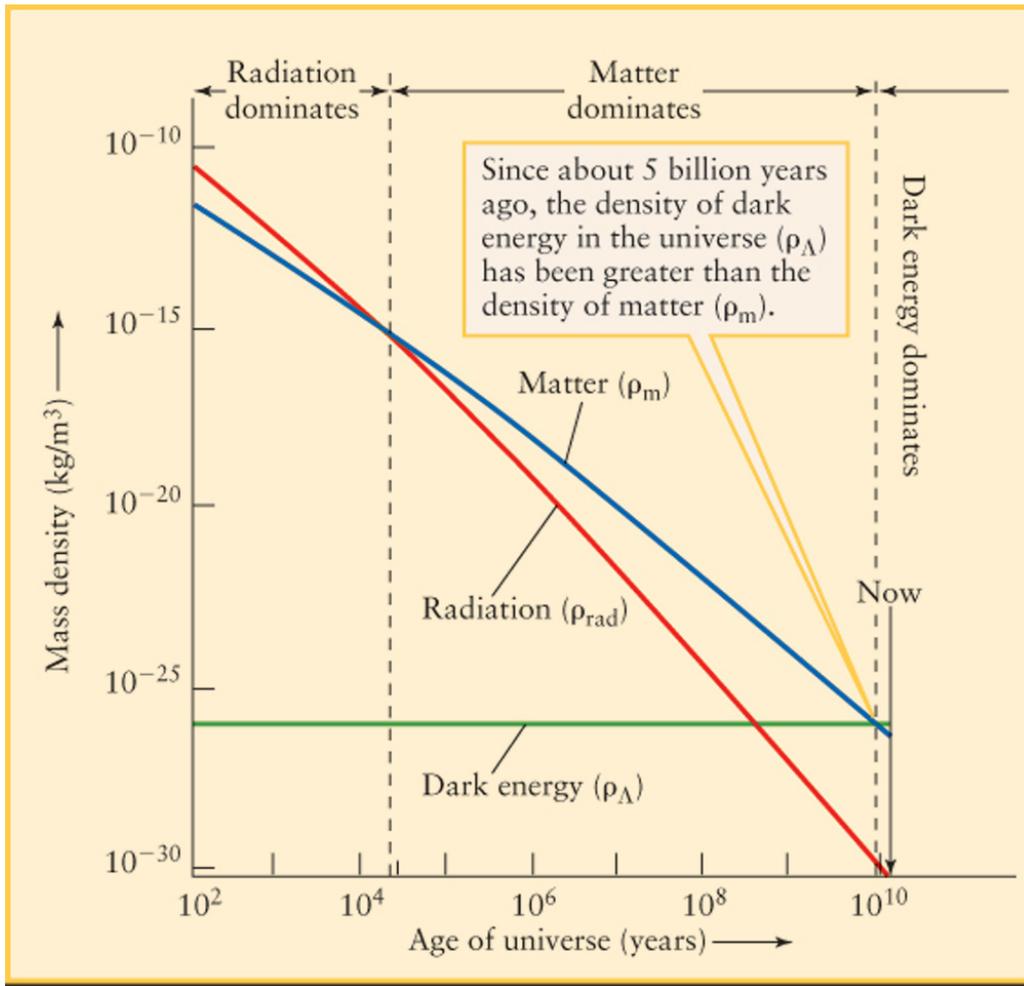


Figure 1.8: *Time-evolution of the Universe energy density components [20].*

$$\begin{cases} \rho_{0,r} \sim 10^{-34} \text{ g/cm}^3 \Rightarrow \Omega_{0,r} \sim 10^{-5} \\ \rho_{0,m} \simeq 0.3 \rho_{0,crit.} \Rightarrow \Omega_{0,m} \simeq 0.3 \\ \rho_{0\Lambda} \simeq 0.7 \rho_{0,crit.} \Rightarrow \Omega_{0\Lambda} \simeq 0.7 \end{cases} .$$

As shown in Figure 1.8 [20], there was a recent phase in the history of the Universe in which  $\rho_\Lambda$  slightly dominated over  $\rho_m$  (*cosmological constant era*). Before, there had been a phase in which  $\rho_m$  dominated (*matter era*) and, even before, a phase in which  $\rho_r$  represented the main contribution to Universe energy density (*radiation era*). At the transition from matter to cosmological constant era, the relevant energy density components had the same value. So, setting  $\rho_\Lambda(z_{eq\Lambda}) = \rho_m(z_{eq\Lambda})$ , one finds

$$\rho_\Lambda(z_{eq\Lambda}) = \rho_{0\Lambda} = \rho_m(z_{eq\Lambda}) = \rho_{0,m}(1 + z_{eq\Lambda})^3 \Rightarrow 1 + z_{eq\Lambda} = \sqrt[3]{\frac{\rho_{0\Lambda}}{\rho_{0,m}}} \quad (1.31)$$

hence by inserting numerical values, one obtains  $z_{eq\Lambda} \simeq 0.3$ . Converting to time, it is possible to realize that the  $\Lambda$  energy density component has become relevant only about 3-4 billion years ago, a relatively recent time if we consider that the Universe is 13.7 billion years old. For this reason, mankind seems to be a privileged observer of this particular transition moment: that is not so likely to happen under a probabilistic point of view.

Another open issue in the  $\Lambda$ CDM model is related to the cosmological constant  $\Lambda$ .  $\rho_{0\Lambda}$  remains constant in time; using Planck satellite data  $\rho_{0\Lambda} \simeq 0.68 \rho_{0,crit.} \simeq 10^{-29} \text{ g/cm}^3$  [21]. So, which is the value of  $\Lambda$ ? In order to get an estimate, we can consider the expression provided by the cosmological Einstein model, so that

$$\Lambda = \frac{4\pi G \rho_{0,crit.}}{c^2} = \frac{4\pi \times 6.67 \times 10^{-8} \times 1.9 \times 10^{-29}}{9 \times 10^{20}} \text{ cm}^{-2} \sim 10^{-56} \text{ cm}^{-2}. \quad (1.32)$$

If one looks at gravitational field equations with a modified stress-energy tensor just like (1.10), the cosmological constant seems to assume the role of a vacuum energy term. De facto, it is not possible to discriminate between a classical cosmological constant theory in which the term  $\frac{\Lambda c^4}{8\pi G} g_{ij}$  is simply added to the stress-energy tensor, and a theory in which the same term is considered as a manifestation of the energy of vacuum (that is known to exist by the observation of Casimir effect): in both cases the additional term affects in some way the space-time curvature and there is no physical chance to distinguish one from the other. Additionally, one could interpret this cosmological constant as the effect of the presence of a delocalized scalar field which dynamically evolves with the Universe expansion. This quantum field would simply be the aforementioned dark energy. The mass of the particle associated to this scalar field would be [22]

$$m_\Lambda = \left(\frac{\hbar}{c}\right) \sqrt{\frac{\Lambda}{3}} \approx 3.5 \times 10^{-66} \text{ g} \quad (1.33)$$

and it is an extremely low value.

Although quantum field theory does not allow to calculate the absolute value of the vacuum energy density, it is possible to calculate how the energy of the vacuum changes at the Universe phase transitions. A phase transition is a moment in the Universe evolution when the

thermodynamical conditions (temperature is taken as reference) reach a value at which the vacuum undergoes a symmetry-breaking transformation, so that new types of interactions and particles are generated. The most relevant phase transitions in the history of the Universe are the following [7]:

$$\left\{ \begin{array}{l} 10^{15} \text{ GeV} \Rightarrow \text{GUT transition} \\ 10^2 \text{ GeV} \Rightarrow \text{electroweak transition} \\ 10^{-1} \text{ GeV} \Rightarrow \text{quarks} \rightarrow \text{hadrons transition} \end{array} \right. \quad (1.34)$$

the energy values are the average particle energy at the time of the transition. Every time a phase transition takes place, the vacuum energy is decreased by a quantity

$$\Delta\rho_{void} = \frac{m^4}{(\hbar/c)^3} \quad (1.35)$$

where  $m$  is the average particle energy at the transition. To compute the vacuum energy at the *Planck time*  $t_P$  ( $\sim 10^{-44}$  s), which is the time before which gravitational quantum effects are relevant, it is sufficient to add to the current value of vacuum energy (which is assumed to be  $\rho_{0,\Lambda}$ ) all the drops due to each transition phase (GUT transition represents the dominant contribution). In natural units one has

$$\rho_{void}(t_P) = \rho_{0,void} + \sum_i \frac{m_i^4}{(\hbar/c)^3} = (10^{-48} + 10^{60}) \text{ GeV}^4 = \rho_{0,void}(1 + 10^{108}). \quad (1.36)$$

It is evident that between the value of vacuum energy density presently associated to  $\Lambda$  and the sum of the energies associated to the phase transitions there are 108 orders of magnitude. One possible solution to this issue is to consider an early and instantaneous phase named *primordial inflation* during which the evolution of the Universe underwent an exponential expansion. Several models have been proposed introducing the presence of a scalar field which would account for the exponential expansion. In particular, the more accepted class of these models assumes a unitary value for the spectral index of the primordial density fluctuations' power spectrum (which is therefore scale-free) and the primordial density fluctuations' distribution is therefore Gaussian. Non-Gaussian terms in the primordial fluctuations' distribution would affect the formation and evolution of cosmic structures: if, for instance, the distribution is positively skewed, then large overdensities become more probable during the early structure formation phases; therefore, greater objects have to be observed at higher redshifts. A conventional way to parametrize the primordial non-Gaussianity in the distribution of density fluctuations is through the gauge-invariant Bardeen's potential  $\Phi$ , which comprises a first order linear Gaussian term and a second order non-linear term accounting for the deviation from Gaussianity:

$$\Phi = \Phi_G + f_{NL} * (\Phi_G^2 - \langle \Phi_G^2 \rangle) \quad (1.37)$$

where the dimensionless parameter  $f_{NL}$  is the weight for the quadratic correction to Gaussian random field  $\Phi_G$  and might depend on the scale [23]. The symbol  $*$  stands for standard convolution and reduces to simple multiplication if  $f_{NL}$  is constant. The Euclid mission will

provide an accurate estimate of  $f_{NL}$  through the measurement of the abundance of cosmological structures.

Does an explanation for the Universe accelerated expansion other than the presence of dark energy exist? It is possible that the late accelerated expansion of the Universe is a gravitational effect due to the presence of inhomogeneities in its matter distribution, but this would require more or less to reject the so-called *Copernican Principle*, which states that our observing point of view is not privileged, thus allowing us to infer that the Universe is homogeneous by measuring its isotropy. There is also the so-called *modified gravity* or *dark gravity* approach. This theory assumes that gravitation does not behave the same on large scales. *Infrared correction* terms are introduced in the gravitational field equations. Setting the Einstein tensor as  $G_{ij} \equiv R_{ij} - \frac{1}{2}g_{ij}R$ , then the field equations become

$$G_{ij} + G_{ij}^{dark} = \frac{8\pi G}{c^4} T_{ij} \quad (1.38)$$

where the  $G_{ij}^{dark}$  term accounts for the infrared gravitation modification [24]. However, a large amount of observational data relative to the solar system and beyond are not accounted for, and in order to adjust the behaviour of General Relativity at low energies, one is forced to introduce a wide range of artificial solutions and instabilities. A way to test General Relativity at large scales is based on the physics of the evolution of cosmic structures. From the Jeans theory, one can derive an expression for the *growth factor* of the primordial density contrast perturbations. By setting  $\delta \equiv \frac{\rho - \bar{\rho}}{\bar{\rho}}$  as the density contrast as a function of time (with respect to the average density  $\bar{\rho}$ ), it is possible to define a cosmic structures' dimensionless growth rate as  $f \equiv \frac{d \ln \delta}{d \ln a}$  [7]. It can be derived that

$$f(\Omega_m, \Omega_\Lambda) \approx \Omega_m^\gamma + \frac{\Omega_\Lambda}{70} \left( 1 + \frac{\Omega_m}{2} \right) \quad (1.39)$$

where  $\gamma$  is an exponent which probes the validity of General Relativity [25]. If GR holds, the  $\gamma$  exponent is close to 0.56. If not, then one has to conclude that General Relativity as we know it, does not work at cosmological scales. However, the presently available data suggest that the measured value for  $\gamma$  is largely compatible with the one predicted by Einstein's theory.

## 1.6 Euclid science objectives

The Euclid mission will be able to improve the accuracy on the measurement of some relevant cosmological parameters. These are:

- $\gamma$ , a General Relativity modification index associated to the growth of primordial density perturbations (GR predicts  $\gamma \approx 0.56$ );
- $m_\nu$ , the sum of neutrino masses (if  $m_\nu \leq 0.1$  eV, the accuracy on the sum of neutrino masses will allow to discriminate between normal and inverted mass hierarchy);
- $f_{NL}$ , non-Gaussianity correction in primordial density fluctuations;

- $w_a$  and  $w_p$ , two parameters characterizing dark energy equation of state,  $p = w(a)\rho c^2 \rightarrow w(a) = w_p + w_a(a_p - a)$ ; their expected values are  $w_p \simeq -0.95$  and  $w_a \simeq 0$ .

Here it is a summarizing table in which all the parameters measured by Euclid are shown (Figure 1.9). Two primary probes, i.e. the observables that allow to derive the cosmological parameters, will be studied by the Euclid mission: gravitational shear, studied through weak lensing effects, and the acoustic oscillations of baryons (BAO, observed through galaxy clustering).

	Modified Gravity	Dark Matter	Initial Conditions	Dark Energy		
Parameter	$\gamma$	$m_\nu/eV$	$f_{NL}$	$w_p$	$w_a$	$FoM$
Euclid Primary	0.010	0.027	5.5	0.015	0.150	430
Euclid All	0.009	0.020	2.0	0.013	0.048	1540
Euclid+Planck	0.007	0.019	2.0	0.007	0.035	4020
Current	0.200	0.580	100	0.100	1.500	$\sim 10$
<b>Improvement Factor</b>	<b>30</b>	<b>30</b>	<b>50</b>	<b>&gt;10</b>	<b>&gt;50</b>	<b>&gt;300</b>

Figure 1.9: Accuracy achievable by the Euclid mission for the measurement of relevant parameters [2]. Euclid Primary: constraints from the combination of galaxy clustering and weak lensing probes. Euclid All: constraints from primary probes and ISW (Integrated Sachs-Wolfe effect). Current: constraints from Rapetti et al. (2009), Komatsu et al. (2010) and Suzuki et al. (2011).  $\gamma$  is the index probing modification of gravity defined in (1.39),  $m_\nu$  is the sum of neutrino masses,  $f_{NL}$  is the weight for non-Gaussianity correction in primordial density fluctuations as seen in (1.37).  $w_\Lambda(z)$  can be approximated by a constant term plus one that accounts for the dynamical nature of dark energy:  $w(a) = w_p + w_a(a_p - a)$ ,  $FoM(\text{figure of merit}) = 1/(\Delta w_p \times \Delta w_a)$ .

### 1.6.1 Gravitational shear

The presence of massive structures in the Universe deflects the trajectory of photons emitted by distant galaxies. This effect results in the deformation of the shape of galaxies emitting photons passing close to a massive structure. Measurements of shape deformations have been used for measuring the mass of clusters of galaxies, and can be used in cosmology for mapping dark matter. Indeed, by observing the Universe at different times, it is possible to measure the correlation between the deformations of the different sources observed at these given epochs in order to trace back the information on the distribution of dark matter present between the sources and the observer. High-resolution cameras are required in order to measure the

mean ellipticity of a group of galaxies in a field and to obtain information on their depth by making measurements at different times (observing in the same field objects located at different redshifts).

Figure 1.10 shows an example of the reconstruction of a dark matter distribution by the HST (*Hubble Space Telescope*) mission. Each line in the image at the center represents the mean ellipticity measurement of 200 galaxies. The length of the lines represents the amplitude of this shearing effect and their orientation is parallel to the principal axis of the shear. The plot on the right illustrates the distribution of dark matter along the line of sight derived by filtering circular structures in the distribution of mean ellipticities. On the left, we see the statistical quantification of dark matter halos on different scales estimated by studying the correlations between the shear effects observed with different lines of sight. The study of the gravitational shear requires very high resolution imaging in the visible wavelength range. It also requires redshift measurements for these galaxies, in order to determine the time at which their light was emitted: this will enable a three-dimensional mapping of the distribution of dark matter in the Universe. Finally, this type of measurement also requires the presence of a minimum density of galaxies to be observed, and a knowledge of the characteristics of the measuring instrument in order to correct systematic effects. Euclid will thus provide a measure of the shape of all galaxies within  $15000 \text{ deg}^2$  of observation with a required galaxy density greater than  $30/\text{arcmin}^2$  in a wide band in the visible ( $550 \text{ nm} < \lambda < 900 \text{ nm}$ ) and a redshift known to better than 5% ( $0 < z < 1.2$  and  $\sigma_z < 0.05(1+z)$ ) [2].

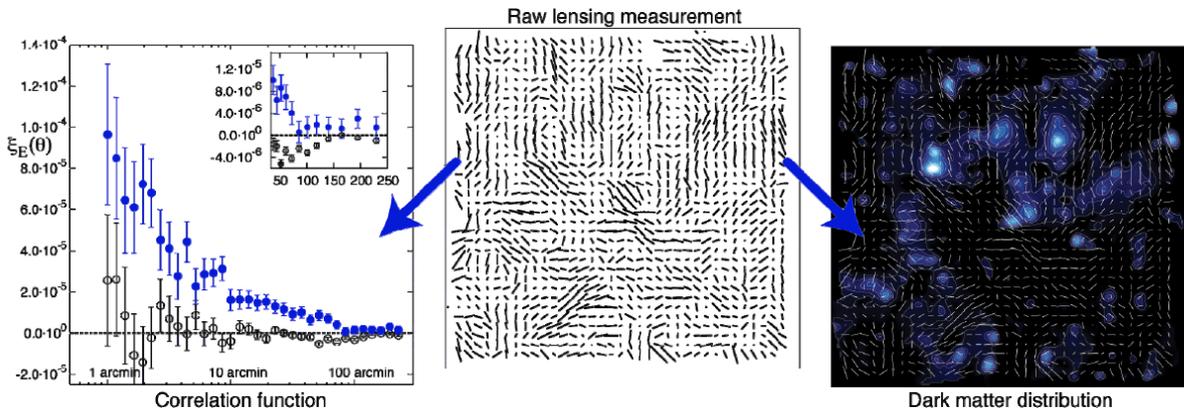


Figure 1.10: *Measurement of the distribution of dark matter from the HST mission by the gravitational weak lensing method [26].*

### 1.6.2 BAO and the Formation of Large Scale Structures

As discussed in §1.2, at a little less than 380000 years after the Big Bang, the Universe was still only a dense plasma of matter and radiation. For structures to form by the action of gravity, small disparities in density should be present in this plasma. According to Guth [27], such disparities arised in the period between  $10^{-35}$  and  $10^{-32}$  seconds after the Big Bang,

when the size of the Universe increased by a factor of at least  $10^{30}$  [28]. The velocity of the expansion conveyed quantum primordial fluctuations to a macroscopic scale leading to small inhomogeneities in the distributions of photons, electrons and baryons. Matter overdensities increased under the action of gravity. As a sound wave is the result of the propagation of an overpressure, in the same way waves generated by the primordial plasma propagate under the effect of the photon pulse transmitted to electrons. Electrons, in turn, carry baryons with them, hence the name of the phenomenon: acoustic oscillation of baryons (BAO, for Baryonic Acoustic Oscillations). After 380000 years, photons interacted much less with baryons and the radiative pressure was no longer sufficient for the acoustic wave to propagate. Therefore, according to their propagation speed, the acoustic waves traveled a certain distance from their starting point leading to material overdensities<sup>2</sup>. Since the recombination epoch, the expansion of the Universe has continued, and the overdensities of matter have moved away from each other while forming structures by the action of gravity. Gravity leads to the formation of the first galaxies. It is possible to find the characteristic distance of the BAOs by studying the spatial distribution of galaxies along the evolution of the Universe (and therefore at different redshifts). This measure allows to determine the rate of expansion of the Universe [30]. The SDSS (Sloan Digital Sky Survey) was the first to provide a large catalog of galaxies (approximately 1 million) located at redshifts  $0 < z < 0.6$  and allowed the measurement of a peak associated with baryonic acoustic oscillations around a spatial frequency of  $100h^{-1}$  Mpc (see Figure 1.11), which indicates the most probable distance between two overdensities.

The observation of baryonic acoustic oscillations requires a precise measurement of the distance separating the galaxies. Spectroscopic observations are needed to measure spectral redshifts of galaxies in order to know their depth in the observed field. The Euclid mission will apply this method to observed galaxies, to cover the  $0.9 < z < 1.8$  redshift area not accessible by ground observations<sup>3</sup> with an estimated precision of  $\sigma_z < 0.001(1 + z)$ . A spectroscopic observation of about 50 million galaxies will be collected. The spectroscopic redshift has to be measured with an accuracy better than  $10^{-3}$  [2].

---

<sup>2</sup>The initial dark matter distribution has remained uniform up to the recombination era, since it does not interact with BAO photons; on the other hand, dark matter is subject to gravitation interaction generated by overdensities of ordinary matter.

<sup>3</sup>Indeed, the measurement of these spectroscopic redshifts requires observing the emission line  $H\alpha$  (the brightest in the spectrum of a galaxy), which at this redshift is shifted in the near infrared wavelength range, that is inaccessible to the ground due to the atmospheric transmission profile.

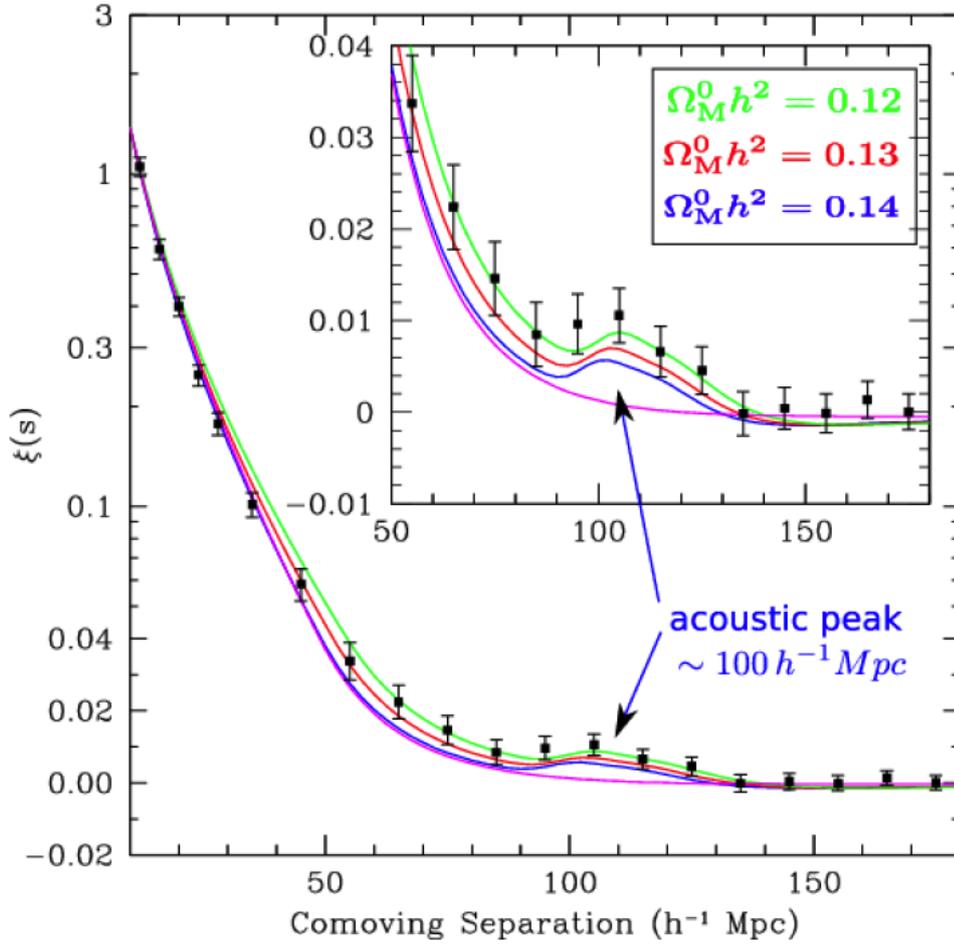


Figure 1.11: Correlation function of the SDSS galaxies: the bump at  $100h^{-1} \text{ Mpc}$  corresponds to the peak of the baryonic acoustic oscillations [29].

---

## Mission architecture

---

Euclid consists in a galaxy survey properly-equipped satellite which will be launched on a Soyuz ST 2.1-B rocket at the end of 2020 from Kourou Space Center (French Guiana). It will be delivered to the Sun-Earth Lagrange point 2 and will conclude its operations within 6 years with an eventual spare activity period of 6 months [31]. The spacecraft is mainly organized into two modules: the Service Module and the Payload Module. The Service Module principally includes all the spacecraft subsystems and the instrument warm electronics; the Payload Module embeds a 1.2 m diameter three-mirror telescope and two instruments: a visible imager (VIS) and a near-infrared spectro-photometer (NISP), Figure 2.1. A single field of view is  $0.5 \text{ deg}^2$ , which corresponds approximately to the size of the moon seen from the Earth. The nominal observation of each field of view is performed through an exposure composed of a sequence of four frames interleaved by in-between dither steps, i.e. small changes in the telescope position ( $\sim 100 \text{ arcsec}$ ) in order to recover the gaps in the image sampling due to the focal plane design of the on-board instruments. Within a frame, the spectroscopic and photometric instruments are exposed to the same portion of the extragalactic sky. Then, in order to avoid disturbance effects from vibrations caused by NISP filter wheel, VIS closes its shutter while NISP completes its photometric imaging procedure. The next field is reached through the execution of a slew (Figure 2.2). The survey program consists of [32]:

- a *Wide Survey*, covering about 36% of the entire sky, corresponding to  $15000 \text{ deg}^2$ . The observation is organized in step-and-stare mode, meaning that the spacecraft will point toward a specific field on the sky and perform a nominal sequence of observations that includes visible imaging, near infrared photometry imaging and near infrared spectroscopic mode; thus, each field on the sky will be covered only once;
- a *Deep Survey*, which will span over about  $40 \text{ deg}^2$  close to the ecliptic poles and will help to calibrate the reconstruction of spectroscopic redshifts and of galaxy ellipticities distribution.

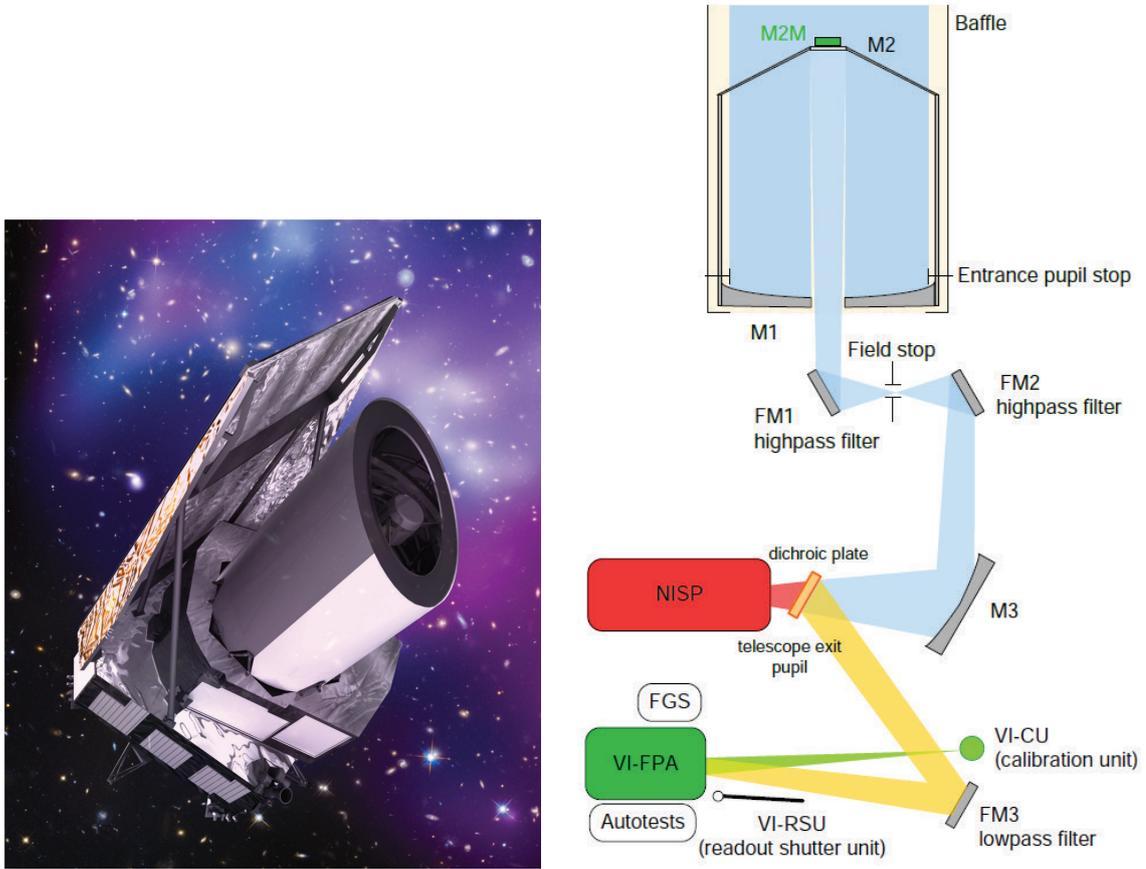


Figure 2.1: *Left: Artistic reconstruction of the Euclid's spacecraft. Right: Sketch of the Euclid's Payload Module. It consists of a three-mirror telescope with a primary of 1.2 m diameter. The two instruments are the visual imager VI-FPA (green), and the NISP instrument (red) which will perform near-infrared photometry and slitless spectroscopy [31].*

The whole survey is based on the observation diagram shown in Figure 2.3 with an overall mapping which systematically avoids the plan of the ecliptic and the galactic plane, areas where the density of stars is a source of important background in the visible and infrared bands. The total number of expected observed galaxies is  $2.5 \times 10^7$  for the galaxy clustering spectroscopic sample and  $1.5 \times 10^9$  for the weak lensing photometric sample [2].

## 2.1 Service Module

The Service Module includes the spacecraft subsystems supporting on-board operations, the warm electronics of the spectro-photometric instruments, the structural interfaces to the Payload Module and the launch vehicle. The Service Module is equipped with a sunshield

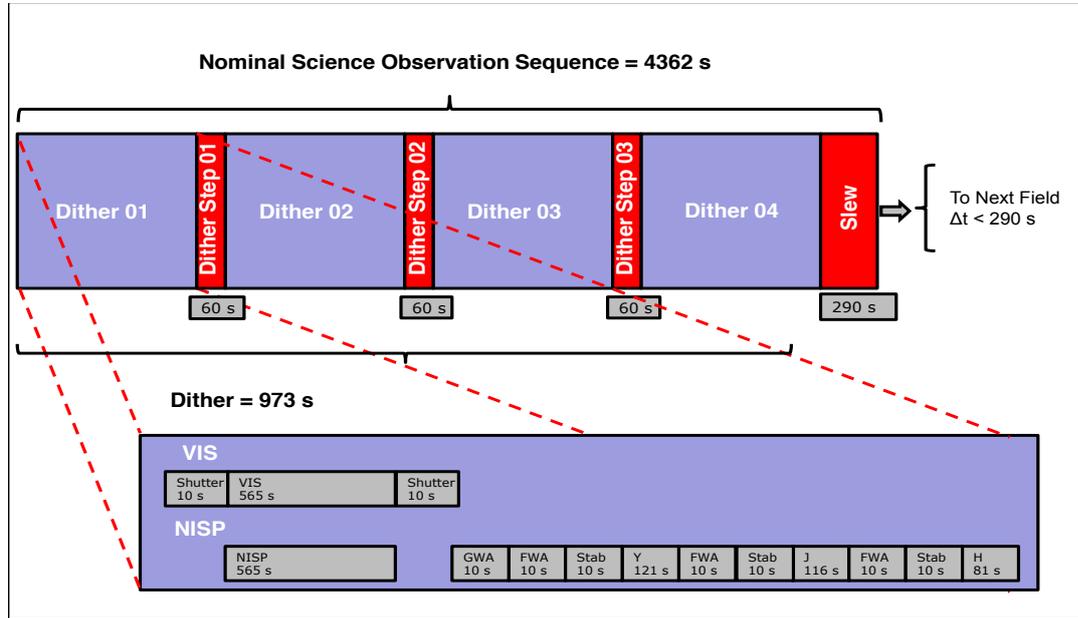


Figure 2.2: *Euclid's nominal observation sequence [31].*

panel to protect the Payload Module from sun illumination and to provide energy for the photovoltaic power supply of the spacecraft. The global spacecraft assembly has dimensions of 4.8 m height and of 3.74 m diameter. The equipments of the Service Module are located to each of its 6 side panels (Figure 2.4): Telemetry, Tracking and Command (TT&C), Attitude and Orbit Control (AOCS), Central Data Management (CDMS) and Electric Power (EPS), payload and Fine Guidance Sensor (FGS) warm electronics. The sunshield consists in a carbon fibre reinforced plastic (CFRP) frame which includes a three-panel photovoltaic assembly on its front side (Figure 2.5). On the top of the sunshield, an optical baffle is installed in order to attenuate to negligible levels the incoming light towards the Payload Module.

The passive thermal control system of the Service Module comprises radiators, Multi-Layer Insulation (MLI) and heaters. The on-board Power Conditioning and Distribution Unit (PCDU) supplies with 28 V on protected lines the spacecraft equipments and instruments electronics, and controls the heaters and the charge and discharge of the battery (which is used only during the launch phase, then substituted by the photovoltaic panel of the sunshield). A centralized on-board computer, the Command and Data Management Unit (CDMU), drives all the communication and data processing flow from and towards the spacecraft and the orbit control unit. The CDMU is a modular unit, made up by the integration of standard processing cores and I/O dedicated interfaces with specific units and devices on-board (see Figure 2.6). The processor module is based on a space-qualified LEON-FT (Fault Tolerant) microprocessor which provides a computational power of 40 MIPS and 5 MFLOPS [31]. *Euclid* will perform scientific exposures and acquire images with a rate of at most 850 Gbit/day [32]. These data have to be stored in a suitable memory budget: the on-board Mass Memory Unit (MMU) provides 4 Tbit

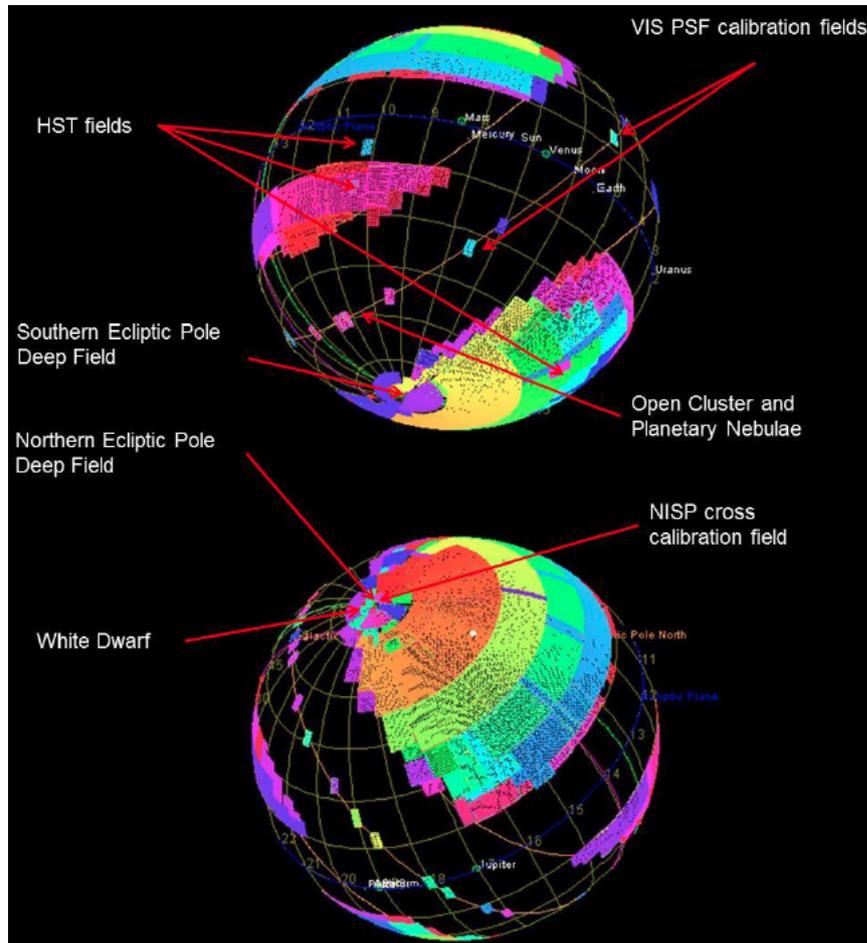


Figure 2.3: *Fields of observation of the Euclid mission in orthogonal projection. The black areas correspond to the plane of the ecliptic and the galactic plane [32].*

which are enough to store 72 hours of scientific data and 20 days of spacecraft housekeeping. As a whole, the MMU will store the housekeeping coming from the subsystems and all the data from the instruments in a two level folders' file system.

Although some spacecraft equipments have dedicated connections, all telecommands and telemetries will be transmitted on two main MIL-1553-STD buses, one linked to the instruments and the mass memory, the other to remaining subsystems. Additionally, the data acquired and elaborated by the instruments are directly delivered to the MMU via a high speed SpaceWire link. The first MIL-1553-STD bus is named "platform bus" and is used for the communication with non-packet Remote Terminals (RT) and FGS: the communication is scheduled in 10 Hz cyclic frames. The other MIL-1553-STD bus, the "science bus", provides a layer protocol to transfer data from subsystems to the MMU: the communication is organized in cyclic frames with 60 Hz rate to maximize the amount of data to be transmitted in a single frame.

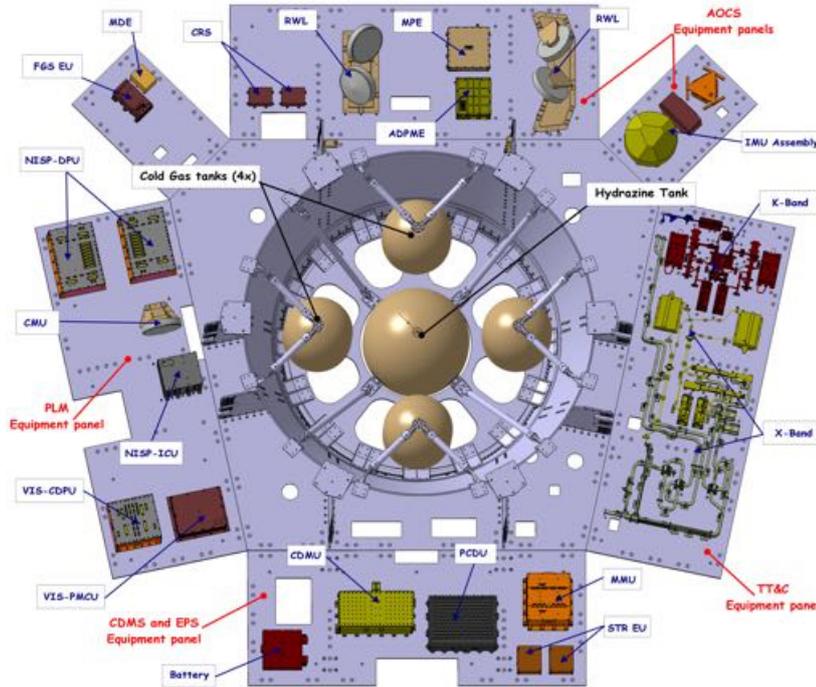


Figure 2.4: *Top view of the Service Module with indication of the position of the various equipments [31].*

The files stored in the MMU are downloaded through the standard CCSDS File Delivery Protocol (CFDP): for the downlink the reliable transfer protocol with acknowledges is used, while for the uplink a simple unreliable transfer is sufficient. The telecommunications are handled via two independent sections: an X-band section used for telecommands and monitoring and a K-band section dedicated to high rate telemetry. CFDP data are transmitted through the X band ( $2.5 \div 3.75$  cm) to guarantee visibility of the download progress to ground regardless of weather conditions, while other data are downloaded via K band ( $1.11 \div 1.67$  cm) to maximize the transmission rate.

The X-band section allows telecommands uplink at 4 kbit/s and 16 kbit/s and housekeeping real time downlink at 2 kbit/s and 26 kbit/s; it uses two X-band transponders, with hot redundant receivers cross-coupled with CDMU TC decoders, and cold redundant transmitters cross-coupled with CDMU TM encoders. The receivers are also cross-coupled with three X-band LGAs (Low-Gain Antennas).

The K-Band section provides science and housekeeping telemetry downlink at two rates: nominal at 73.85 Mbit/s and reduced at 36.92 Mbit/s to allow an extra link margin in case of bad weather. The K-Band section includes two cold redundant modulators, directly linked to the MMU for memory files' download, and is coupled with a 70 cm diameter K-band HGA (High-Gain Antenna).

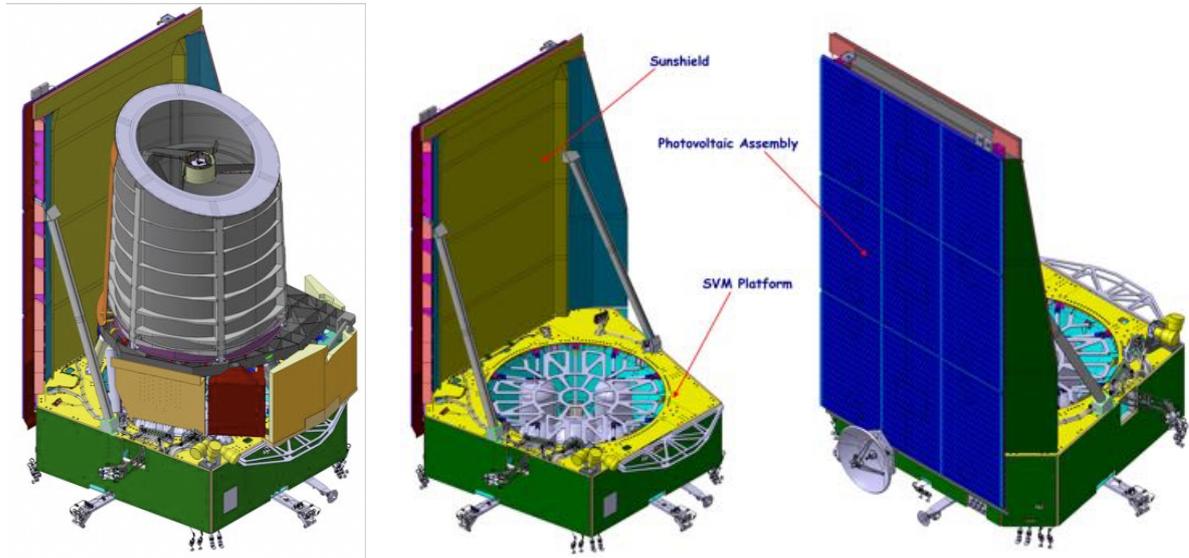


Figure 2.5: *Left: front view of Euclid's spacecraft. Center and Right: overview of the Service Module with sunshield position highlighted [31].*

## 2.2 Payload Module

The Payload Module is based on an anastigmatic three-mirror Korsch telescope [34] made of Silicon Carbide (SiC) which directs the incoming light to VIS and NISP instruments (see Figure 2.1). The primary mirror (M1) with a diameter of 1.2 m collects the light and returns it to the secondary mirror (M2) equipped with a three degrees of freedom mechanism for focus and tilt corrections (M2M). The M2 mirror directs the beam, which passes through a filter, to the last mirror (M3) in charge of delivering the luminous flux to the instruments of the satellite. Between mirrors M2 and M3, the two folding mirrors<sup>1</sup> FM1 and FM2 are placed to direct the light beam in the plane of the baseplate. A third folding mirror FM3 is used to increase the efficiency of radiative area for VIS. The transmission of the light flux towards the instruments is done by a dichroic plate located at the exit pupil of the telescope, which separates the light in 50/50 shares.

The telescope field of view is of  $1.25 \times 0.727 \text{ deg}^2$  and the collecting area is about  $1 \text{ m}^2$ , so that the flux provided is sufficient for the operation of the two on-board instruments. The Payload Module includes also the FGS (Fine Guidance Sensor), which represents the Attitude and Orbit Control pointing information reference. All the electronics is mounted on the Service Module (with the exception of the FGS and focal plane's proximity electronics) so that the possible thermal effects affecting the Payload Module are reduced to a minimum. The Payload Module is divided into two volumes by the baseplate (Figure 2.7):

- the *front cavity*, comprising the primary and secondary mirrors with the M2 mirror refo-

<sup>1</sup>A folding mirror is a mirror used to change the direction of the reflected light beam.

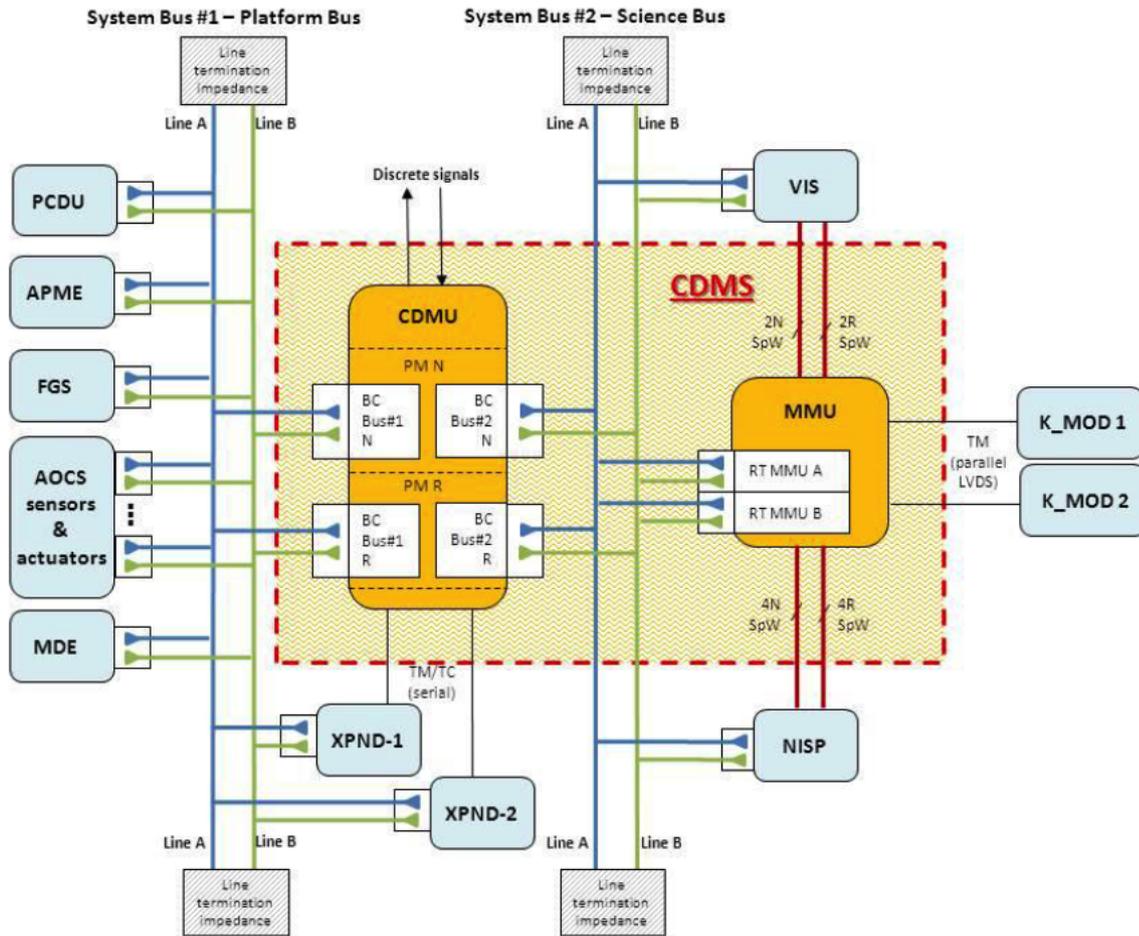


Figure 2.6: Command and Data Management System interfaces [33].

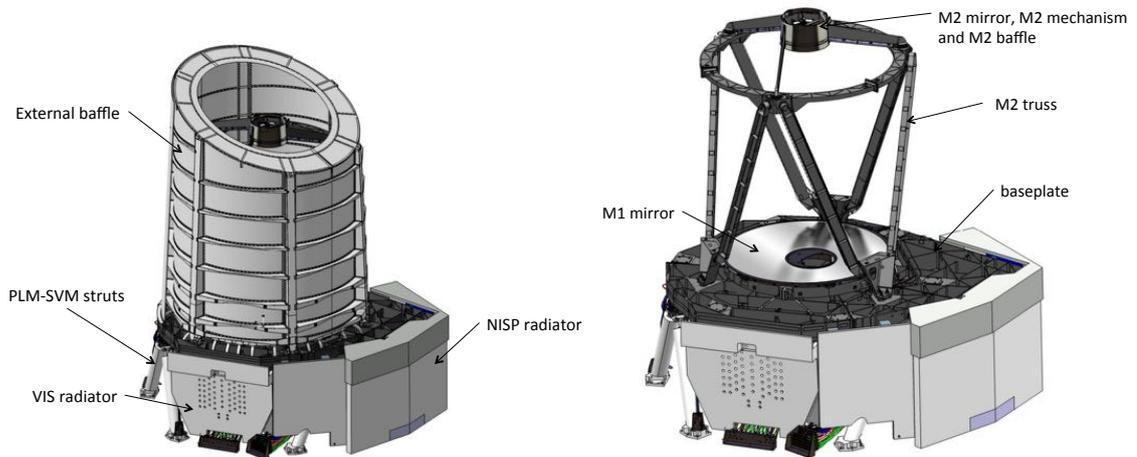


Figure 2.7: *Left: external front view of the Payload Module. Right: external front view of the Payload Module without the external baffle [31].*

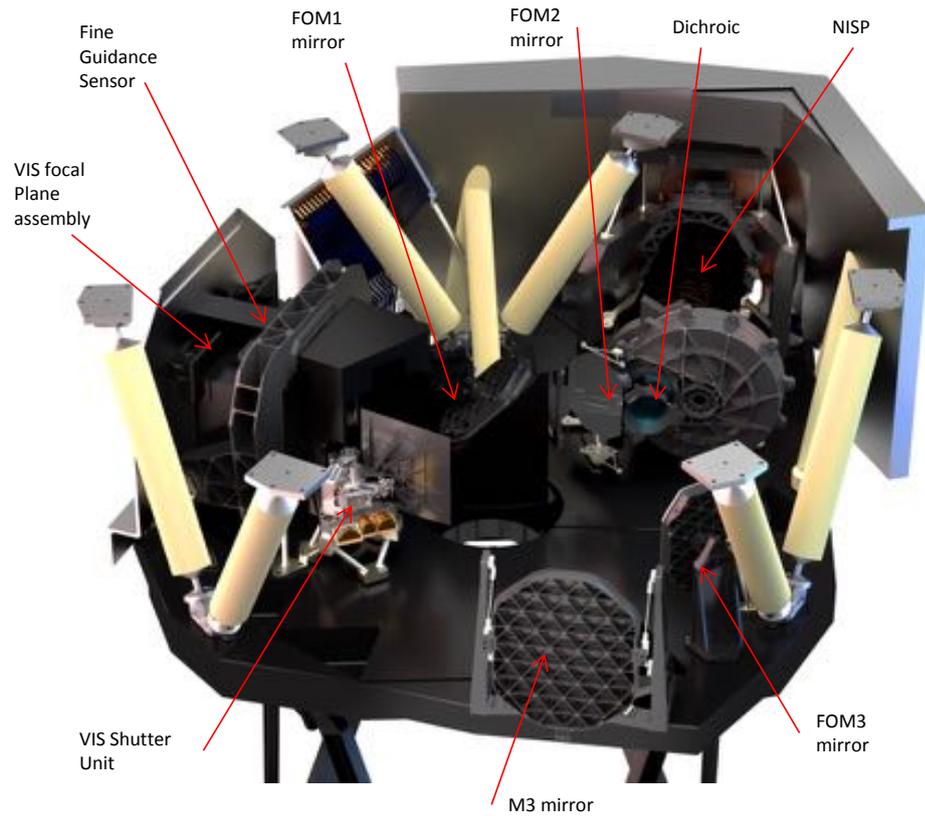


Figure 2.8: *Rear view of the Payload Module showing the instrument cavity content [31].*

cusing apparatus, and thermally insulated by a specific baffle;

- the *instrument cavity*, including the folding mirrors, the dichroic splitter, the FGS, the NISP and VIS instruments, the shutter and calibration unit of VIS (Figure 2.8).

The mechanical architecture of the Payload Module is conceived in order to fit the selected thermal architecture for the telescope and the instruments, passively maintained at the same temperature. The thermal passive design is totally based on that of the GAIA mission, since it provided good thermal stability with minimum power consumption [35]. The equilibrium temperature at which the telescope is kept is about 130 K. The required heating power during nominal operations is quite low ( $\sim 140$  W), since only local heating capacity is needed to maintain instruments interfaces' temperature at a stable and prescribed level. The instruments are kept at cold temperatures (135 K) and in order to limit possible disturbance from the telescope structure, Silicon Carbide offers the minimum thermo-elastic responsiveness (the coefficient of thermal expansion is  $0.4 \mu\text{m}/\text{m}/\text{K}$  [31]). Moreover, the heat leaks to cold instruments are minimized, and the Payload Module becomes a heat sink for all the units contained with the exceptions of VIS-FPA electronics and NISP, which dissipate heat through dedicated out-looking radiators. The use of specific harness and conductive and radiative thermal interfaces guarantees the reduction of heat leaks from the Service Module and the sunshield.

### 2.2.1 The VIS instrument

The VIS (visible imager) instrument is designed to produce images with high resolution in the  $550 \text{ nm} < \lambda < 900 \text{ nm}$  wavelength range, in order to measure the deformations of the observed galaxies. VIS has a field of view that is more than 100 times larger than that of the Hubble telescope [36]. The astrophysical objects are detected at a  $10\sigma$  SNR limit for extended sources of AB magnitude 24.5. Besides, the VIS instrument is composed of several subsystems (see Figure 2.10):

- the *focal plane array* (FPA), equipped with  $6 \times 6$  CCDs operated at 150 K (Figure 2.9). Each CCD is composed of  $4\text{k} \times 4\text{k}$  pixels, so that the total detector matrix is composed of more than 600 million pixels. With a pixel plate scale of 0.10 arcsec ( $12 \mu\text{m}$ ), the geometric field of view is  $> 0.5 \text{ deg}^2$  (corresponding to  $29 \text{ cm} \times 29 \text{ cm}$ );
- the *calibration unit*, consisting of an integral sphere equipped with LEDs, providing uniform illumination over the entire focal plane;
- the *shutter mechanism*, placed upstream of the focal plane: as soon as an observation ends, it is closed to avoid stray light during data processing;
- two *electronics units*, one for the control and processing of instrument data, the other for the control of the mechanisms and the power allocated to the instrument; the central data processing unit receives data from CCDs and compresses the resulting large image ( $24\text{k} \times 24\text{k}$ ) in about 250 seconds; the power and mechanisms control unit switches the shutter and the calibration unit on/off.

In Table 2.1 VIS subsystems functions are summarised.

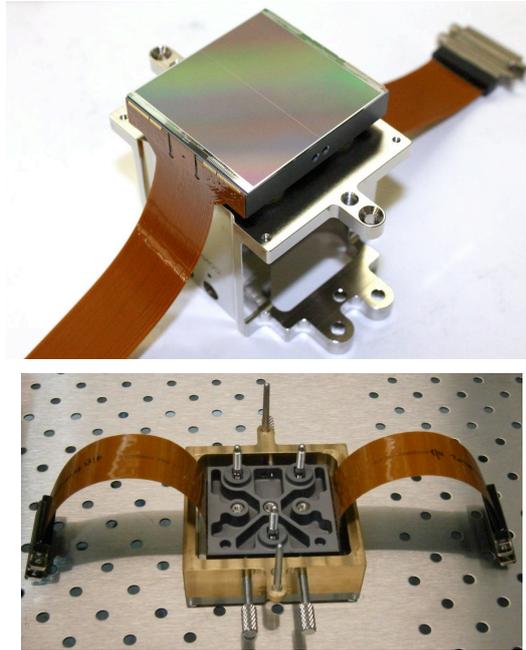


Figure 2.9: The CCD273-84 to be used in VIS. A view of the active surface (top) shows the arrangement of the flexible connectors. The rear surface (bottom) shows the SiC packaging.

Name	UNIT	Function
<b>VI-FPA</b>	VIS Focal Plane Assembly	Detection of visible light for imaging
<b>VI-SU</b>	VIS Shutter	Close VIS optical path for read out Close VIS optical path for dark calibration
<b>VI-CU</b>	VIS Calibration Unit	Illuminate the FPA with Flat Field for calibration
<b>VI-CDPU</b>	Control and Data Processing Unit	Control Instrument Perform data processing Interface with Spacecraft for data handling
<b>VI-PMCU</b>	Power and Mechanism Control Unit	Control Units
<b>VI-FH</b>	Flight Harness	Connection of units

Table 2.1: Functional description of VIS units.

### 2.2.2 The NISP instrument

The NISP is a spectro-photometer performing slitless spectroscopy and imaging photometry in the near-infrared (NIR) wavelength band. In order to observe galaxies between  $z = 0.9$  and  $z = 1.8$ , photometric and spectroscopic observation are performed. In Figure 2.11 the main subsystems of the instrument are shown.

- two *filter wheels* (NI-FWA/GWA), Figure 2.11, one hosting the various optical filters

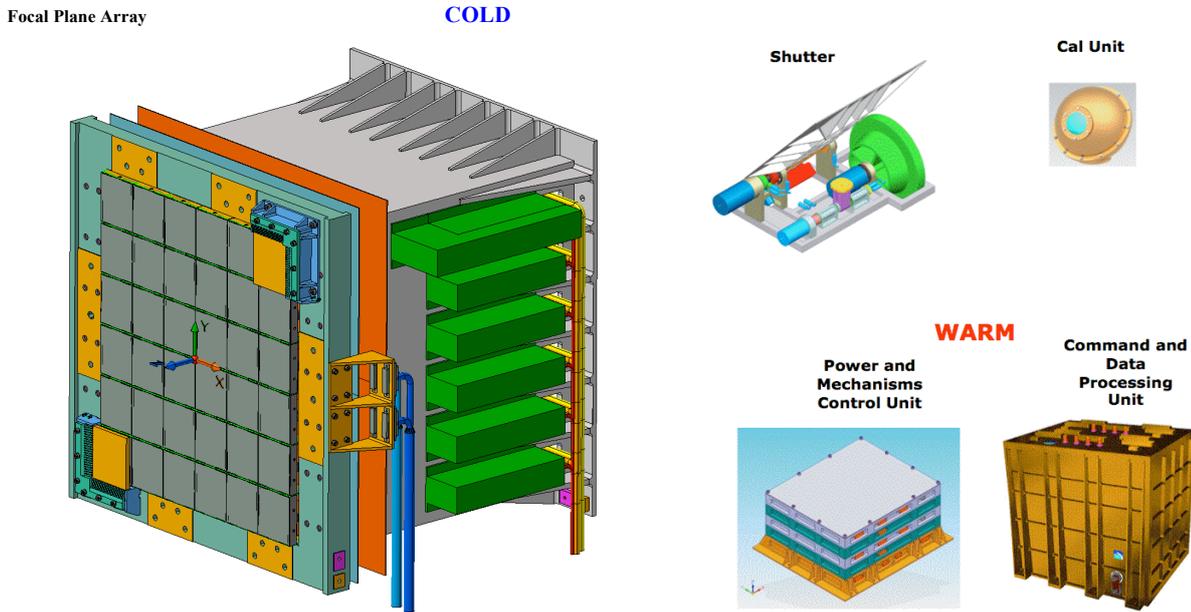


Figure 2.10: Units comprising VIS: Power and Mechanisms Control Unit and Command and Data Processing Unit in the bottom right are in the Service Module, while the Focal Plane Array, the Shutter and the Calibration Unit are in the Payload Module.

Name	UNIT	Function
NI-OMA	NISP Opto-Mechanical Assembly	This holds the optical elements and the Focal Plane Array
NI-GWA	NISP Grism Wheel Assembly	This holds the four dispersing elements for the spectroscopic mode and it allows them to be placed in the optical beam.
NI-FWA	NISP Filter Wheel Assembly	This holds the three filters for the photometric mode and it allows them to be placed in the optical beam. It provides also a closed and open position.
NI-CU	NISP Calibration Unit	This injects calibration signal in the optical beam for calibration purposes
NI-DS	NISP Detector system	This system provides detection of the NIR signal in photometric and spectroscopic mode
NI-WE	NISP Warm Electronics	This is composed of the NI-DPU and NI-ICU (see below).
NI-DPU	NISP Data Processing Units	This provides data compression and packeting as well as the interface to S/C Mass Memory
NI-ICU	NISP Instrument Control Unit	This controls the instrument, powers and controls mechanisms, provides instrument thermal control, and the command interface with NI-DPU.

Table 2.2: The main elements which NISP comprises.

required for photometric observation mode (in Y, J and H bands) and the other the grisms (the arrangement of the filters/grisms is shown in the bottom of Figure 2.11);

- a focal plane (NI-DS), Figure 2.11, equipped with an array of  $4 \times 4$  HgCdTe detectors. The detectors will be operational at a temperature between 80 and 100 K; this kind of

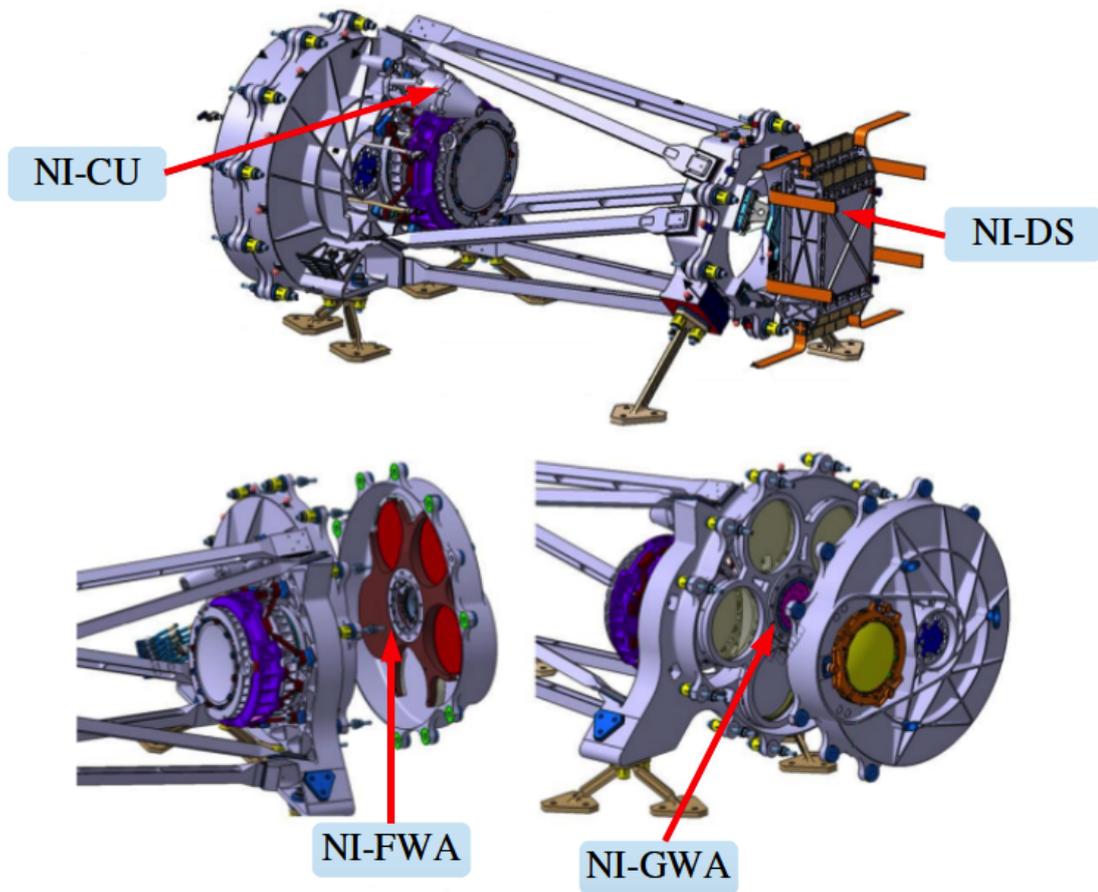


Figure 2.11: 3D sketch of the NISP instrument with the NISP Filter Wheel Assembly and Grism Wheel Assembly, Detector System and Calibration Unit.

detectors is named H2RG. Each detector consists of a matrix of  $2048 \times 2048$  pixels, thus the whole focal plane is a matrix of more than 60 million pixels, covering a surface of  $16 \text{ cm} \times 16 \text{ cm}$  (Figure 2.12). Each pixel has a size of 0.3 arcsec, so that the overall field of view is of  $0.5 \text{ deg}^2$ . The detector wavelength cut-off is at  $2.3 \mu\text{m}$ . The sensor chip system of the detectors is made of a sensor chip assembly, an ASIC readout electronics, cryogenic flexi cable connections linking the sensors to the readout units, a telemetry and telecommand router based on SpaceWire Link technology, a set of interconnections between the ASIC and the SpaceWire Link and a thermal controlling and monitoring system.

- the *electronics unit* (NI-WE), located in the Service Module. It allows the synchronous acquisition of the focal plane detectors, as well as in-flight data pre-processing.
- a *calibration unit* (NI-CU), located close to the filter wheel.

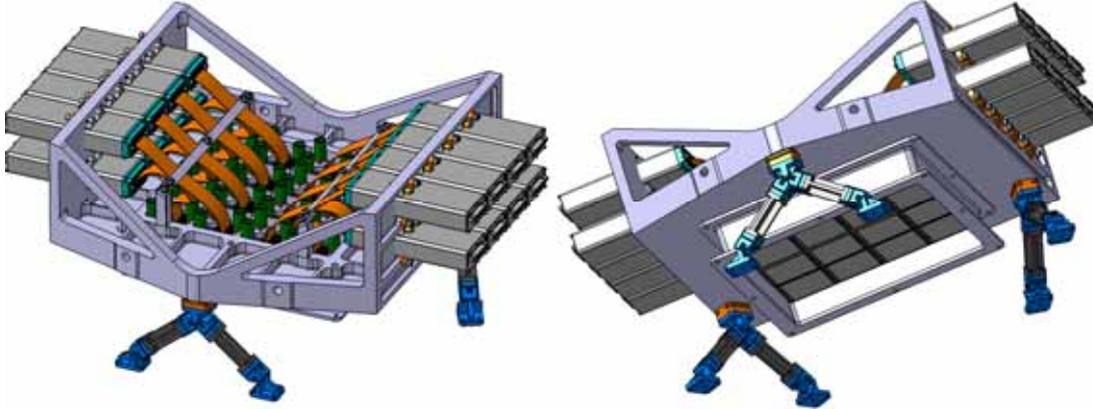


Figure 2.12: *NI-DS layout overview: the cryogenic support structure is made of SiC and holds the mosaic of detectors, two orthogonal walls embed the readout electronics.*

In Table 2.2 NISP subsystems functions are summarised.

### 2.2.3 NISP instrumental performances

NISP observations in photometric mode are made using a Filter Wheel Array (NI-FWA, Figure 2.13) which provides five positions: three positions are occupied by filters each 10 mm thick and with a diameter of 120 mm in the Y, J and H bands. The other two positions are equipped with an opening and a hood to close the instrument. The overall filter passband is  $920 \text{ nm} < \lambda < 2000 \text{ nm}$ . The photometric detection limit for  $5\sigma$  SNR point sources is of 24 AB magnitude.

Slitless spectroscopy is Euclid's mean to investigate galaxy clustering. The redshifted  $H\alpha$  emission line (656.3 nm at rest) is used to measure the spectrum and redshift of galaxies in the  $0.9 < z < 1.8$  range and  $0.9 \mu\text{m} < \lambda < 1.825 \mu\text{m}$  wavelength range. For an average source of 0.5 arcsec the minimum detected flux ( $3.5\sigma$  SNR) is  $2 \times 10^{-16} \text{ erg}/(\text{s} \cdot \text{cm}^2)$  [37]. In the spectroscopic mode a grism<sup>2</sup> wheel (NI-GWA) with five positions is used: four are occupied by grisms with different passbands, one with blue coating and the others with red coating, and the last position is an open position (Figure 2.13). The rotation of the two wheels (NI-FWA and NI-GWA) is ensured by a cold mechanism that can operate in vacuum from 300 to 20 K.

### 2.2.4 NISP observation sequence

There are a total of seven observation modes for the NISP instrument:

1. GWA grism 1 with FWA in open position;
2. GWA grism 2 with FWA in open position;

<sup>2</sup>A grism is a transmission grating mounted on a prism, so that light diffraction at a certain wavelength gets compensated by the prism deviation.

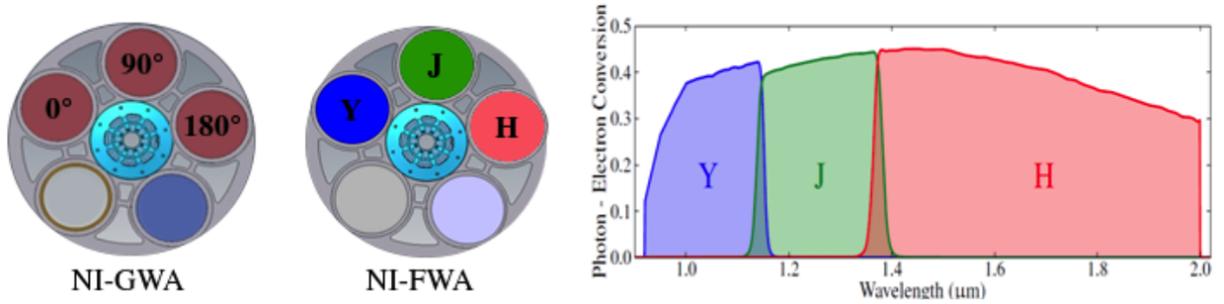


Figure 2.13: *Left: NISP Filter Wheel Assembly and Grism Wheel Assembly and their five positions. Right: Profiles of the NIR filters of the NISP instrument.*

3. GWA grism 3 with FWA in open position;
4. open GWA with FWA in position Y;
5. open GWA with FWA in position J;
6. open GWA with FWA in position H;
7. FWA closed.

The first three modes are spectroscopic modes, the following three are photometric modes and the last one is the “dark” mode for the measurement of the background noise. The combination of observation modes into an observation sequence is the so-called “science sequence”. The “science sequence” illustrated in Figure 2.14 can be divided into four cycles: each cycle is the succession of a spectroscopic mode and three photometry modes, at the end of which a “dithering” is performed, which consists in a slight shift of the field of view of the telescope to eliminate sampling effects. The two instruments, NISP and VIS, are active at the same time during the observation sequences, the VIS observing during the NISP spectroscopic sequences. Once this observation sequence has been completed, the telescope moves to another field (*slew*). During this time, the instrument is closed (observation mode 7) and a background noise measurement is made. This observation technique allows to carry out Euclid data record within the 6 years mission, but implies that it is not possible to return to observe the same field during the survey. An accurate knowledge of the instrument is therefore mandatory.

### 2.2.5 NISP electronics architecture

The NISP Warm Electronics (NI-WE), is located in the Service Module at a temperature of more than 240 K. It comprises a set of main functional units: the NISP Detector System, the NISP Data Processing Unit and the NISP Instrument Control Unit (Figure 2.15). The Opto-Mechanical Assembly, containing the optical elements, is located in the Payload Module at cryogenic temperature. It is controlled by NI-ICU which commands also the operations of the

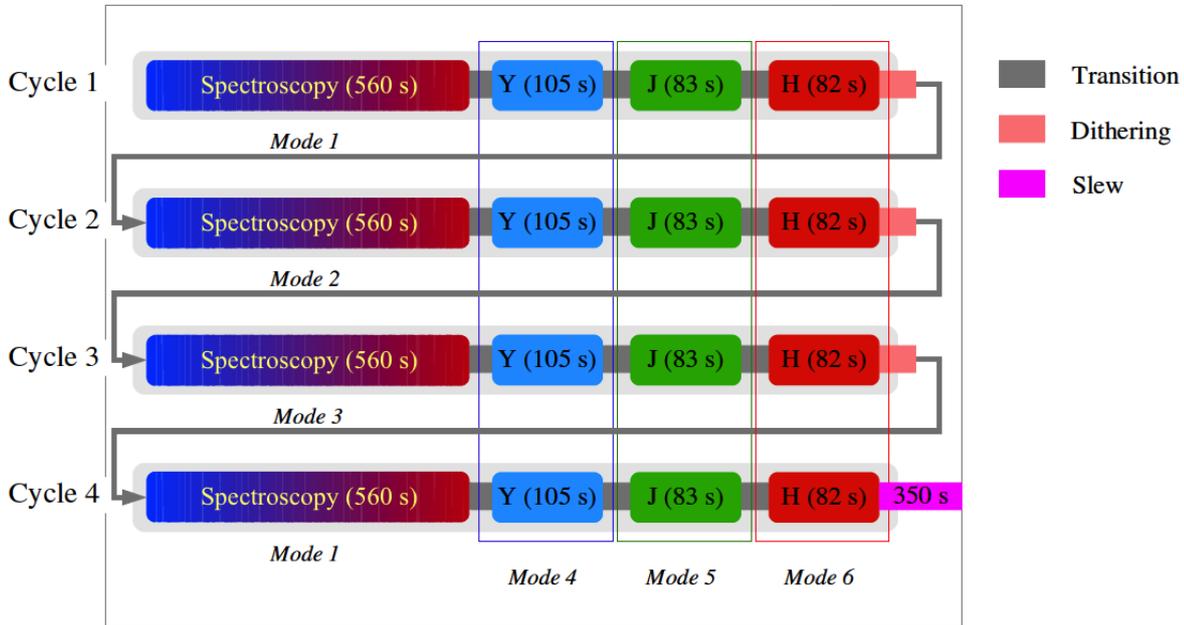


Figure 2.14: Representation of the observation cycles forming the “science sequence” of the NISP instrument for a field of  $0.5 \text{ deg}^2$ . The spectroscopic modes (modes 1, 2 and 3) are at the beginning of each cycle, followed by the photometric modes (modes 4, 5, 6). During the slew period, separating two science sequences, the instrument is closed (mode 7).

other NISP subsystems. The processing of data acquired by NI-DS is performed on-board by NI-DPU. The Detector Control Unit DPU handles the interface with the ASICs. The DCU is in charge of the pixel data pre-processing according to the readout modes and the calculation of the slopes with the necessary corrections. The NI-DPU performs science data compression and transmission to the spacecraft Mass Memory, manages the communication with the NI-ICU, including commands and housekeeping data, and maintains the file configuration for booting the ASICs. It is also configurable for detector calibration readout mode. The NISP setup comprises two DPUs, as shown in Figure 2.16.

The NI-ICU supervises all the NISP functionalities and provides the interface to the S/C (spacecraft) control system. It includes specific control electronics for the NI-FWA, the NI-GWA, the NI-CU, the temperature control of the NI-OMA and the NI-DS, and preserves the housekeeping data of the NI-WE. The electronics has been conceived in order to keep a single H2RG channel as independent as possible from the others, thus reducing the possibility of generating single-point failures.

The NISP software comprises the NI-DPU software, for instrument data management, and the NI-ICU software, dedicated to the handling of the S/C platform interface and instrument commanding: this includes communication with S/C and related subsystems, and health status checks. Periodically checking the housekeeping parameter values, NI-ICU can predispose the

NI-CDMU to partly or entirely switch off the NISP instrument.

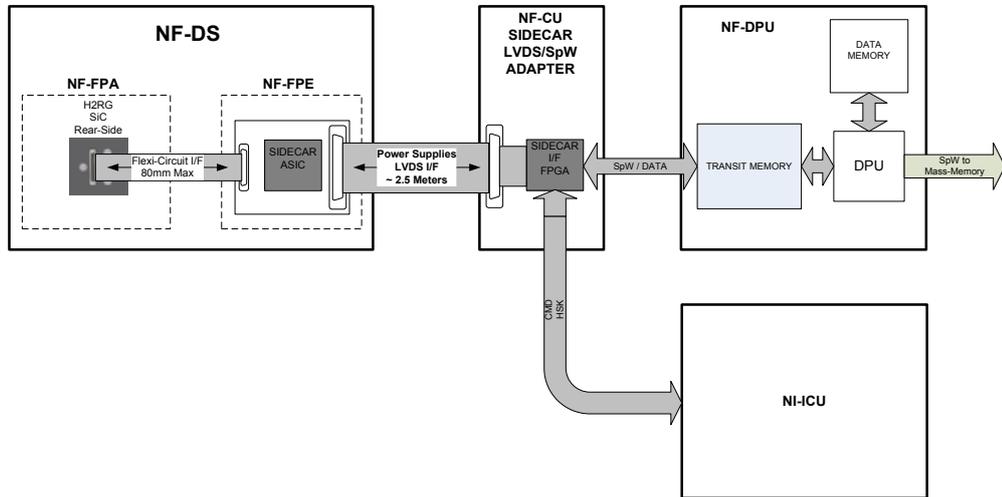


Figure 2.15: Block diagram showing a single NISP detection channel (one out of 16) [2].

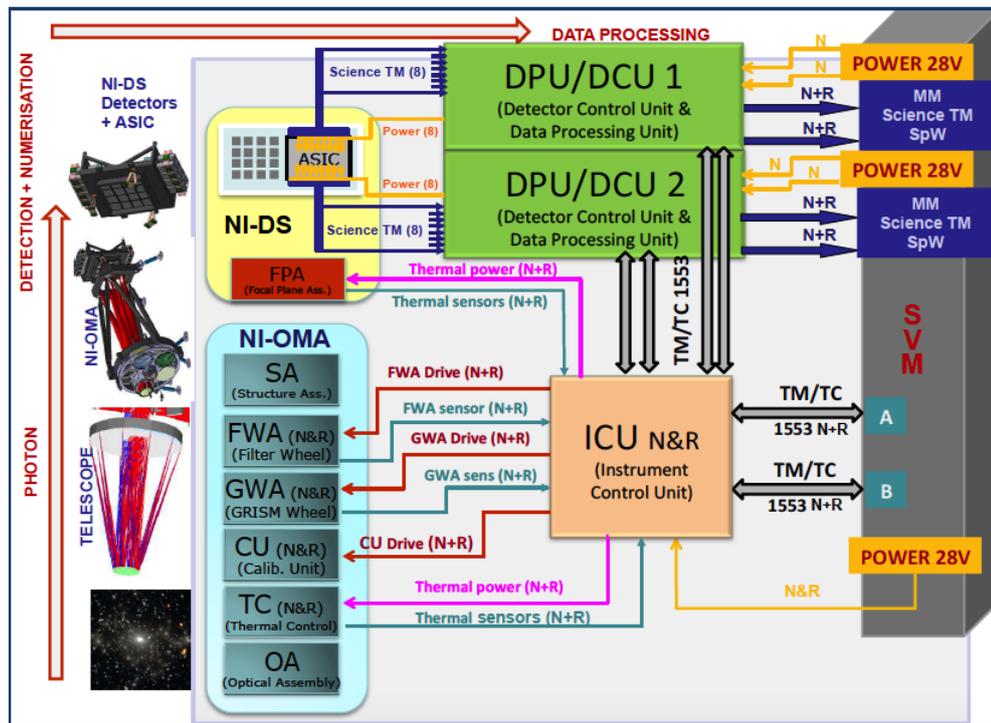


Figure 2.16: Schematic overview of NISP components [37].

In this chapter the software simulating the counterparts of the ICU communication interface (Spacecraft and DPU) is described. The aim was to provide a software capable of simulating the communication interface between the S/C CDMU and the NI-ICU (i.e. telemetry and telecommand exchange between the spacecraft and the instrument control unit). In the following a description of the NI-ICU and its ASW is provided.

### 3.1 The NI-ICU

The Euclid NISP Instrument Control Unit (NI-ICU) is based on 3 modules (Figure 3.1):

1. the *Low Voltage Power Supply* module (LVPS), providing DC/DC converters and 1553 transceivers for the connections with the NI-DPUs;
2. the *Central Data Processing Unit* (CDPU), including an MDPA (Multi-DSP  $\mu$ -Processor Architecture) ASIC (Application-Specific Integrated Circuit), 1553 transceivers for the high level MIL-STD-1553 interface with the S/C and a test connector;
3. the *Driving & Analog Support* module (DAS), handling all the analogue acquisition and driving electronics, comprising heaters, temperature sensors, calibration LEDs and filter and grism wheels.

The CDPU is the NI-ICU central processing unit hosting a LEON2-FT (Fault Tolerant) processor, embedded in the MDPA ASIC together with an RTAX (Radiation Tolerant) FPGA (Figure 3.2); the latter is used to integrate the MDPA standard functionalities (e.g. implementing timers, watchdog and On-Board Time timer) and to make the CDPU board interact with the DAS by means of a SPI (Serial Peripheral Interface) link that goes through the NI-ICU backplane. The CDPU allows loading, debugging, and monitoring an application through the Debug Support Unit (DSU) of the LEON2 processor.

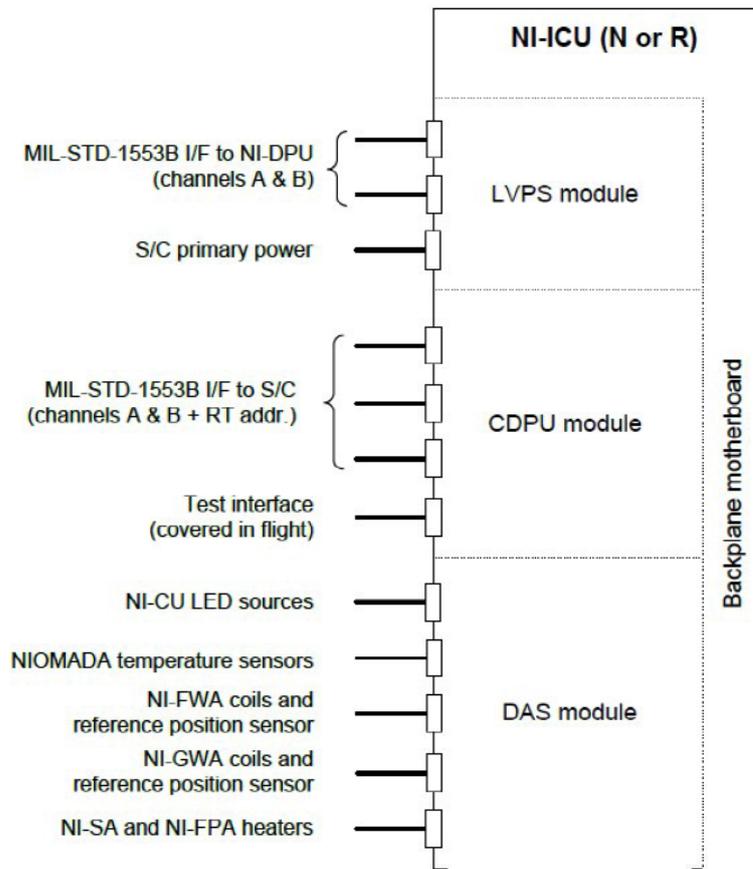


Figure 3.1: Schematic overview of the NI-ICU architecture [38].

## 3.2 The NI-ICU ASW

The NI-ICU ASW is based on the space-qualified RTEMS (<https://www.rtems.org>) real-time operating system (version 4.8) provided and tailored by EDISOFT Defence & Aerospace Technologies [39]. In Figure 3.3 a block diagram of the executive environment of the Application Software is shown, including RTEMS kernel and abstraction layers for hardware functions. The ICU-ASW is designed to handle:

- the satellite/platform communication interface
- the NI-ICU/NI-DPUs communication interface
- the NI-OMA subsystems (through the DAS board)
- all the instrument commanding and monitoring functionalities

and is in charge of several functions:

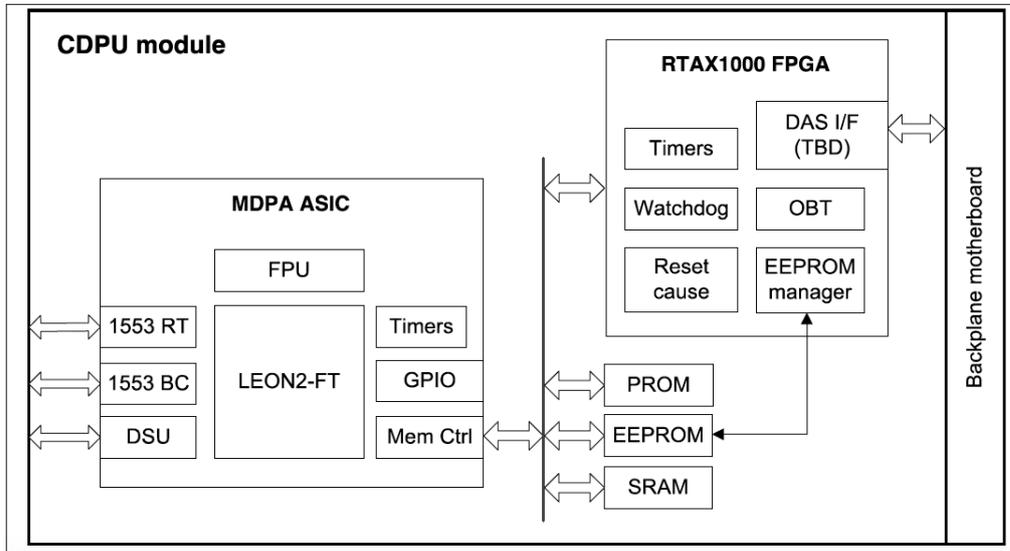


Figure 3.2: *Hardware Functions for the NI-ICU CDPU module.*

- management of the operating modes of NISP instrument;
- telemetry (TM) and telecommand (TC) exchange with S/C CDMU in accordance with ECSS (*European Cooperation for Space Standardization*) standards, which prescribe the implementation of MIL-STD-1553 communication protocol and of TM/TC PUS (*Packet Utilization Standard*) packet formatting;
- distribution of TCs to NI-DPUs via 1553 link and to NISP electronics connected to NI-ICU through the DAS (such as filter and grism wheels, calibration unit and thermal control);
- global monitoring of TM produced by all NISP instrument modules;
- management of the software maintenance through patch and dump functionalities<sup>1</sup>;
- management of On-Board Time (OBT) and TM time tagging in order to keep synchronization with the S/C and to transmit related timing information to the NI-DPUs, so that NISP instrument internal synchronizations are preserved;
- control of the enabling/disabling and of the level of current absorption of the calibration unit;
- management of the reference position sensor of the filter wheel;

<sup>1</sup>The operation of patching files in the EEPROM of the NI-ICU CDPU can be done only through the NI-ICU Boot Software (see §3.2.3).



Figure 3.3: Block diagram illustrating the NI-ICU ASW executive environment.

- control of the stability of NI-OMA and of NI-FPA cold plate condition through temperature sensors and heaters;
- execution of FDIR (*Failure Detection, Identification and Recovery*) processes.

The functions of NI-ICU specific hardware (processor resources, memories, 1553 IP cores, SPI interface and other functions of the NI-ICU CDPU board) are managed by an abstraction layer implemented into NI-ICU Driver Software (DSW) consisting of libraries of low-level functions.

### 3.2.1 ICU-ASW Architectural design

The NI-ICU ASW consists of a series of software layers:

- a hardware layer, which is represented by the drivers managing the NI-ICU-connected devices under RTEMS;
- an operating system abstraction layer, allowing software development regardless of the used real-time operating system;

- a low-level services layer (PUS services, device management, TC handling, TM packetization);
- a high-level components layer.

Figure 3.4 shows a scheme representing the conceptual structure of the ASW, with its main tasks, their logical interconnections and the related interfaces (either hardware, software or both). The NI-ICU ASW is designed to handle the following categories of configuration data:

- configuration of the NI-ICU itself, which is installed on-board together with a redundant unit: since the wheel operation profile is different between nominal and redundant unit (the home sensor is displaced by 180 degrees), it is necessary to distinguish between the two configurations;
- configuration of the NISP instrument: it comprises the nominal observation sequence settings, the NI-DPU readout mode selection and the configuration of the wheels and of the calibration unit;

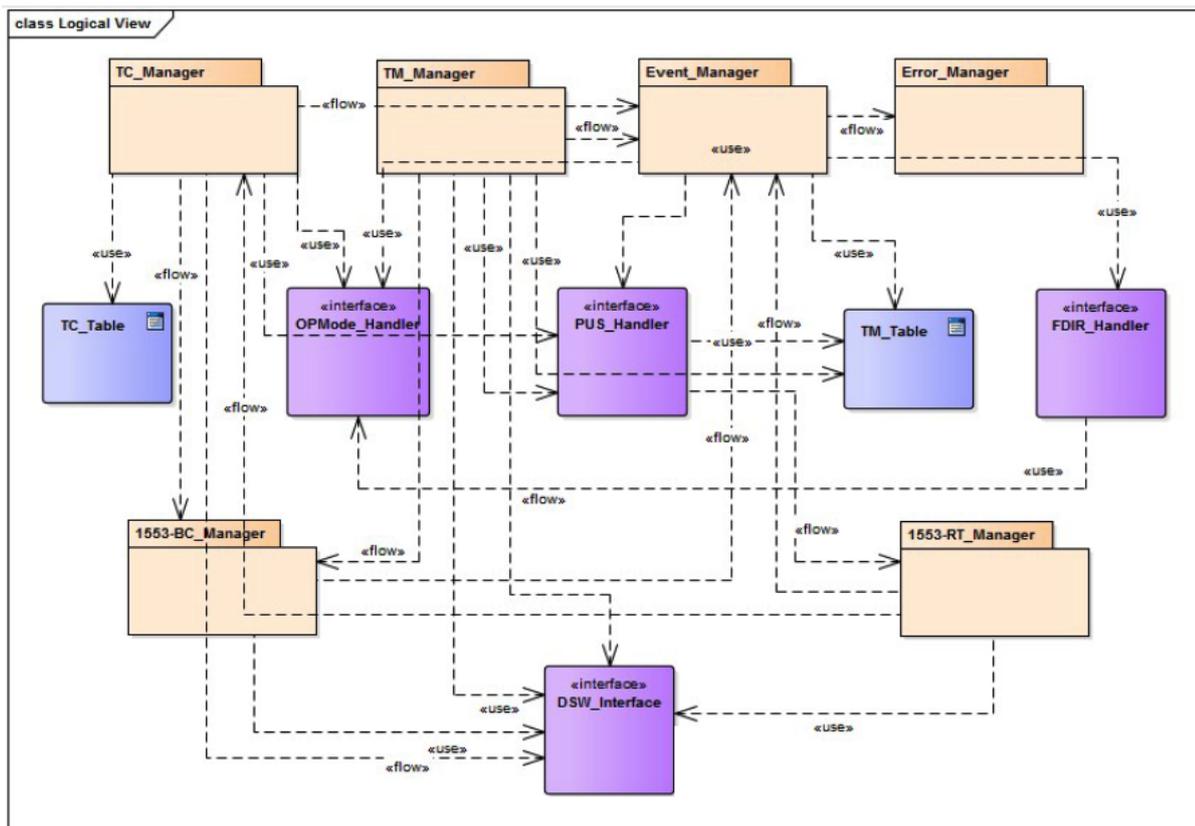


Figure 3.4: Block diagram illustrating the NI-ICU ASW logical structure.

- configuration of the NI-DPUs and of related electronics (e.g. firmware files).

The main software components of the ICU-ASW are:

- the *command interpreter* and *sequencer*: it waits for the arrival of a TC and when one TC arrives, its formal validity is checked comparing with the corresponding hardcoded TC prototype; then transmits the validated and decoded TC to the sequencer, which in turn may execute an on-board procedure related to the command, or deliver the command to the device in charge of its execution; the command interpreter is also responsible for the implementation of the various PUS services (see §3.2.2);
- the *time manager*: it receives the time information provided by the S/C via the corresponding 1553 interface and synchronizes the system time. It produces periodic TM packets with information on the status of the synchronization;
- *TM collection* and *formatting* component: it is in charge of receiving and monitoring housekeeping (HK) data sent by the NI-ICU-connected devices; it produces TM packets formatted in TM tables and sends them to the S/C, recalling the intervention of the *event manager* in case of out-of-limits parameter values;
- the *event manager*: it forwards events or alarms through the 1553 interface and executes autonomous procedures if required from the analysis of HK parameters; moreover, it can trigger the execution of FDIR processes;
- the *error manager*: it manages software errors, providing error codes and sending events to the event manager;
- the *S/C 1553 interface manager*: it consists of a 1553 Remote Terminal (RT, essentially the slave unit for the 1553 communication protocol) which polls the 1553 buffer waiting for new data. When a TC is received, it is sent to the command interpreter; at the same time. It reads incoming TM tables and transmits them to the S/C following a specific bus profile;
- the *NI-DPU 1553 interface manager*: it consists of a 1553 Bus Controller (BC, essentially the master unit for the 1553 communication protocol) which handles the communication with the NI-DPUs on the related 1553 connections: it sends TCs formatted according to an internal packetization protocol and receives TM packets which are transmitted in turn towards the S/C. Additionally, it sends the time information to the NI-DPUs, in order to maintain the system internal synchronization;
- *TC table*: it contains static information about TC formats and limits over their parameters to allow TC packets validation;
- *TM table*: it contains information coming from the various NI-ICU subsystems transmitted on the 1553 link according to the time profile defined for the corresponding TM group.

### 3.2.2 PUS services

The NI-ICU ASW, as well as the VIS-CDPU Application Software, uses the PUS standard for the TC/TM communication with the S/C CDMU. The PUS standard is a set of monitoring and control services prescribed by the European Cooperation for Space Standardization (ECSS). The Euclid PUS standard provides only the services necessary for the mission (see Table 3.1).

<b>PUS service #</b>	<b>Service description</b>
1	Telecommand Verification
3	Periodic Reporting
5	Event Reporting
6	Memory Management
8	Function Management (via NISP specific TCs)
9	Time Management
17	Connection Test

Table 3.1: Description of Euclid PUS standard services.

### 3.2.3 Operational flow

The NI-ICU ASW is launched by the NI-ICU Boot Software (NI-ICU BSW). The NI-ICU BSW manages the startup procedure, checks the integrity of the NI-ICU CDPU EEPROM (containing 2 images of the NI-ICU ASW) and launches the selected ASW image after receiving the related TC. The ICU-ASW is designed in such a way that the TM/TC exchange protocol with the S/C is initialized through a dedicated reset procedure. The protocol comprises two counters, one for TM packets and the other for TCs. The TM counter is incremented every time the NI-ICU sends telemetry to the S/C, while, on the contrary, the TC counter is increased every time the S/C transmits a telecommand to the NI-ICU. After the protocol reset procedure has been accomplished, the counters for TM and TCs must be equal to zero, and so the NI-ICU ASW gets ready to receive TCs from the S/C CDMU. These TCs belong either to a timeline file stored on the S/C (for default standard operations) or to a schedule received from ground in real time (for example when anomalies occur and troubleshooting is necessary). The TCs are received by the NI-ICU and then are either executed in the NI-ICU itself (if they directly involve the NI-ICU action or the activation of one of the NI-ICU connected devices), or are delivered to the NI-DPUs for the commands that involve the scientific data acquisition chain. The timeline analysis shows that the normal rate of received TCs is quite low (typically 1/s). The structure of the NISP instrument is such that some commands (in particular those dispatched to the NI-DPUs) have long execution time. This is true in particular for the exposure commands. In general, a TC is processed according to the schemes shown in Figure 3.5 or in Figure 3.6. Two types of TM packets are foreseen to be exchanged with the S/C in order to inform on the TC

processing progress:

- TC acceptance report (PUS 1,1 for successful validation or PUS 1,2 for validation failure);
- TC completion report (PUS 1,7 for successful execution or PUS 1,8 for execution failure).

No report is foreseen for the TC start of execution as by PUS 1,3 or PUS 1,4, neither on TC execution progress (PUS 1,5 or PUS 1,6). The TC validation is always done by the NI-ICU, according to the TC syntax check, operation mode and parameter check (see Figure 3.5). The TC is executed, i.e. the start of execution packet is issued, if the involved subsystem is a NI-ICU device (for instance the NI-GWA). According to the device, the TC successful completion report is issued based on measurements (for instance for the NI-CU) or on the basis of the logic state of the device (for the wheels).

The commands dispatched to the NI-DPUs is managed in a different way since the interface for TC/TM with the NI-DPUs is more complex. The TC acceptance is performed in a similar way as the other commands. The TC completion report, successful or failed, is based on the NI-DPU HK information, whose acquisition is part of the processing sequence of TCs targeted to NI-DPUs. A selection of parameters collected by the NI-ICU ASW upon receiving HK information from the various subsystems, is regularly monitored and if values exceeding

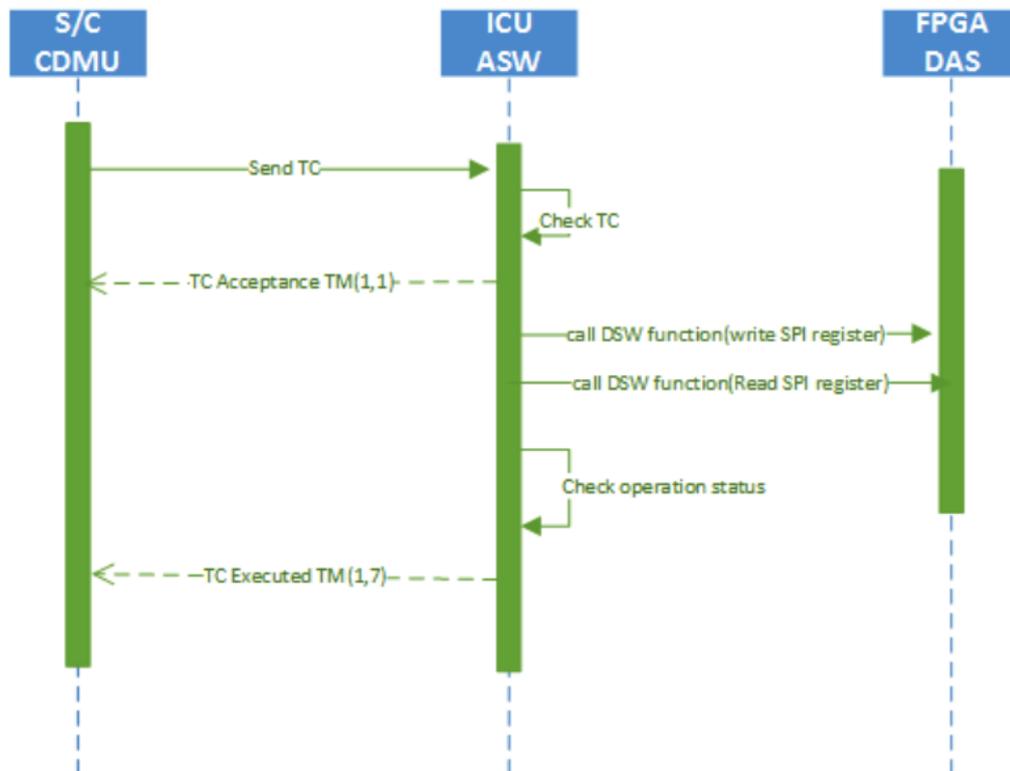


Figure 3.5: Execution diagram of a TC addressing a DAS connected device.

thresholds are found, a flag or an event is generated; in case of alarms, FDIR algorithms are triggered. Since some of the commands (in particular the commands for the wheel motion) will be time-tagged (depending on the operation mode), time synchronization is of relevance, but, taking into account the rate of commands and the other timing requirements, a level of 0.1 seconds would be acceptable, and it is well within the capabilities of standard 1553 time synchronization mechanism. An example of a sequence diagram for an accepted TC is shown in Figure 3.6. In the worst case the time between two subsequent TCs is 2 seconds.

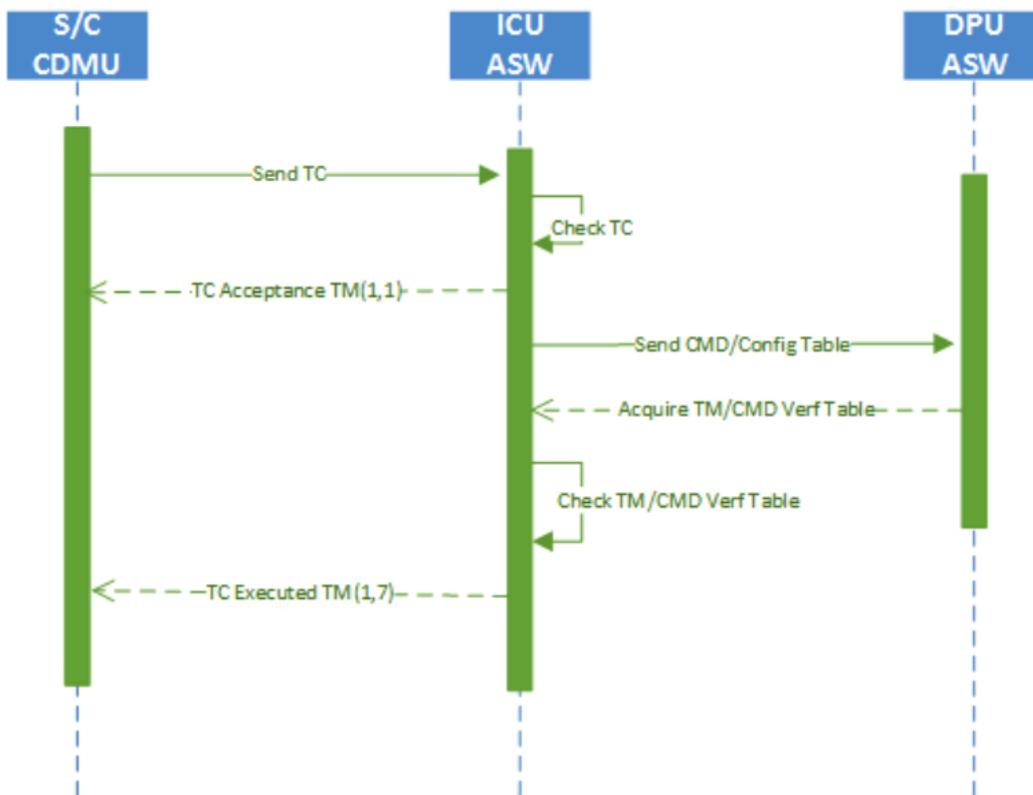


Figure 3.6: Execution diagram of a TC for a NI-DPU.

### 3.2.4 Operating modes and mode transitions

In this section we focus on the NISP instrument operating modes, managed by the NI-ICU ASW, in charge of handling the permitted transitions between these modes. The operating modes are described in detail in the mission official document NISP IOCD (Instrument Operations Concept Document [40]). An operating mode is mainly defined by the set of hardware and software configurations for the subsystems which are enabled and active. With the exceptions of NI-ICU and NI-DPUs, all NISP units can assume only “on” and “off” operational states. The NI-ICU and NI-DPUs are able to perform transitions among a large set of opera-

tion modes on the basis of specific rules managed by the ICU and DPU Application Softwares. Figure 3.7 shows a schematic overview of NISP operating modes and allowed transitions, with the indication of the mechanism used to implement the transition itself, via telecommand, automated action or both. The following list highlights the characteristics of currently defined NISP operation modes:

- *off*: all NISP instrument units are switched off;
- *boot*: only NI-ICU is on and NI-ICU BSW is running;
- *startup*: all NISP instrument units could be in principle switched on and the full HK is made available; NI-ICU boot sequence is successfully accomplished and the other subsys-

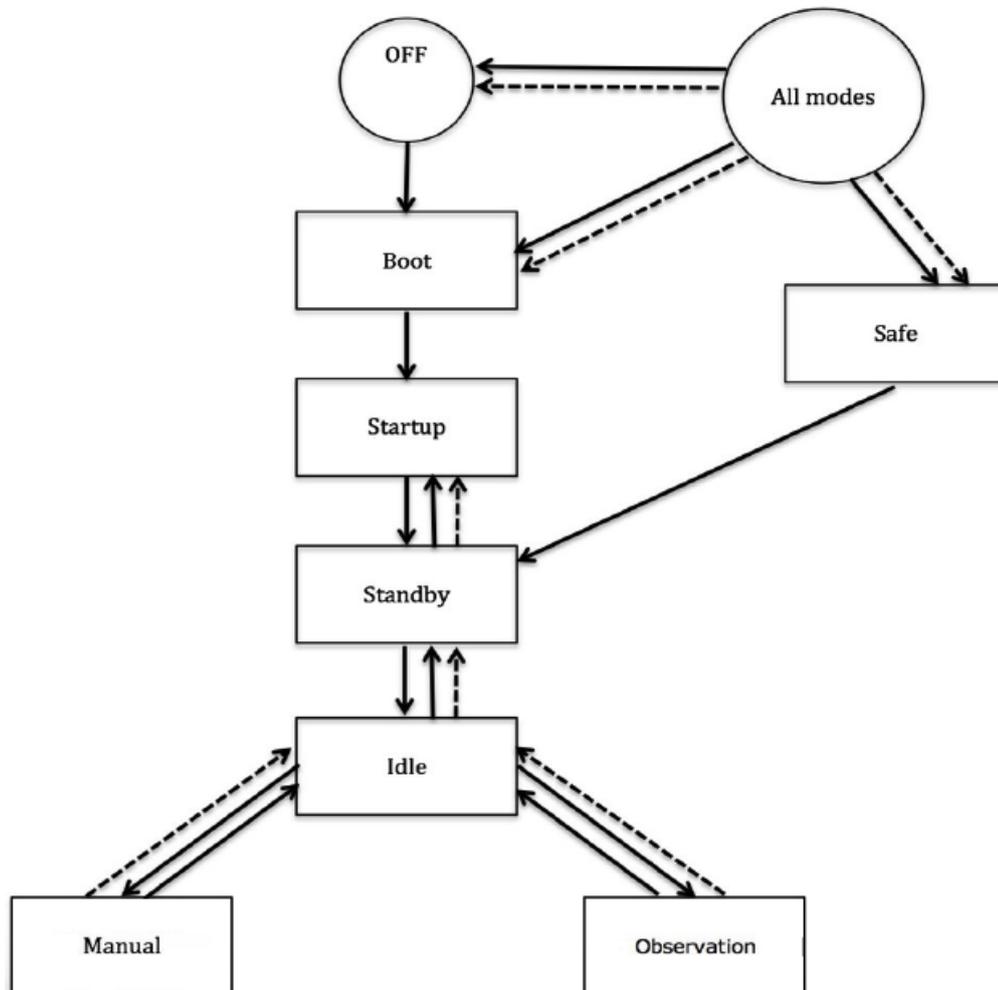


Figure 3.7: *NISP operating modes and transitions. Solid lines indicate commanded transition, dashed lines refers to autonomous transitions.*

tems could be activated;

- *standby*: NI-ICU and NI-DPUs are on; NI-ICU can receive TCs from the S/C and NI-DPUs have already been booted;
- *idle*: this mode is expected to be used when minor failures occur;
- *manual*: only NI-ICU must be on; this mode guarantees the maximum flexibility in order for the NISP instrument to be commanded by ground;
- *observation*: all NISP units are switched on and the instrument is in nominal operational state;
- *safe*: this mode is expected to be used when major failures occur.

### 3.2.5 The test equipment for NI-ICU ASW interface

The ASW of the NI-ICU is developed mainly on a bread board model (EBB) representative of the ICU-CDPU module. The Bologna INFN team set up a specific equipment to test the communication through the MIL-STD-1553 interface of the NI-ICU. In particular, I took care of the implementation of S/C and NI-DPU simulators to be used extensively for the development of the ICU-Application Software.

As a partial result of my activity, I wrote a user manual in order to explain how to set up the development environment required to load, run and debug the ICU-ASW on the EBB (see Appendix B).

The test equipment is a combination of hardware and software elements to implement the counterparts of the EBB interfaces:

- an *SPI link* to connect with the DAS Board;
- a platform MIL-STD-1553 bus (*science bus*) to communicate with the S/C;
- a MIL-STD-1553 bus (*internal bus*) to communicate with the NI-DPUs.

These counterparts provide the commanding, monitoring and stimulation of the NI-ICU and allow the debugging and verification of the NI-ICU ASW. The EBB has also two more interfaces, for programming and debugging purposes:

- a *Test Interface*, which is a serial line to communicate with the LEON2 DSU allowing non-intrusive debugging of the complete target system through the Gaisler GRMON tool [41];
- a *JTAG port*, which is used to program the FPGA of the CDPU board with Actel Flash-Pro4 tool [42].

The CDPU is connected to a secondary board (*setup board*) included in the EBB delivery. The setup board implements functions associated to the CDPU test interface (reset, LVDS-to-RS232 DSU serial line adapter) and to the CDPU backplane (power supply, reset, SPI

link). The setup board hosts physically a second (external) FPGA module (DLP-HS-FPGA3), accessible through an USB interface. A diagram of the EBB interfaces is shown in Figure 3.8, and a picture of the final setup is shown in Figure 3.9.

The commercial power supply unit (PSU) used to power the EBB setup board has to implement a foldback protection<sup>2</sup>. The expected consumption for the EBB setup with a power supply positive voltage between 36 V and 75 V (the nominal value maintained during working sessions is 48 V) is:

- setup + CDPU in standby: 85-95 mA at 48 V (4.3 W);
- while memory test is executed: 110-130 mA at 48 V (5.7 W);
- expected maximum consumption: < 500 mA at 48 V (< 24 W).

The simulation of the DAS module is performed by the so-called *SecoiaSim*. The SECOIA (Serial Control Interface Asic) is an ASIC that implements a wide range of interfacing functions commonly used in instrument control units of space missions. In the NI-ICU, SECOIA will be used for controlling the motors, the calibration LEDs, the heaters and the ADC (*Analog-to-Digital Converter*) that will acquire the analogue telemetry of the NISP instrument. The DAS module of the NI-ICU will actually include the SECOIA, plus all the associated driving and acquisition electronics. The SecoiaSim is a SECOIA hardware simulator which implements the same SECOIA interface; it runs on the FPGA module hosted by the EBB setup board and is connected to the workstation through a USB interface.

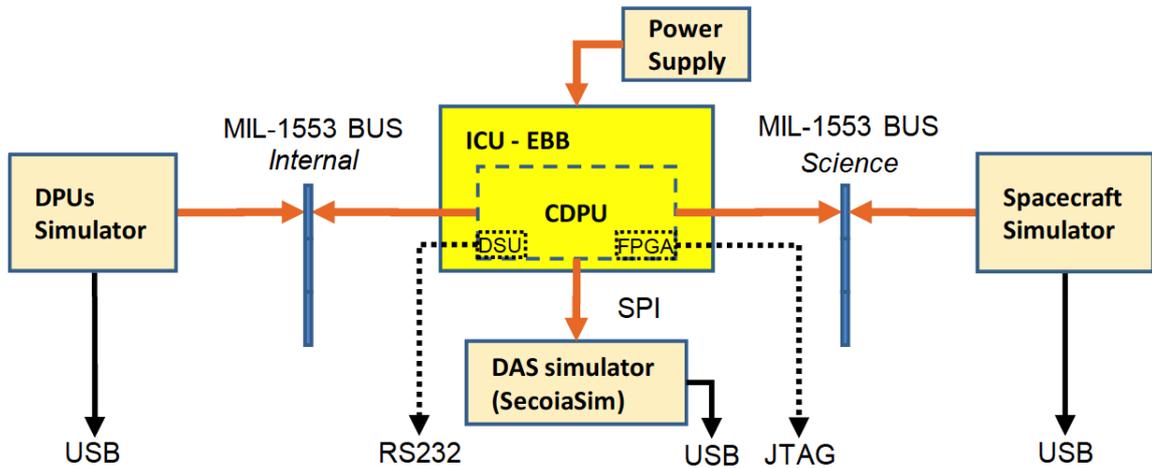


Figure 3.8: NI-ICU breadboard (EBB) interfaces and test equipment connected for the application software development.

<sup>2</sup>Foldback is a current limiting feature of power supplies which reduces both the output voltage and current to well below the standard operating limits in case of current overload. The main purpose of foldback is to keep the output transistors within their safe power dissipation threshold.

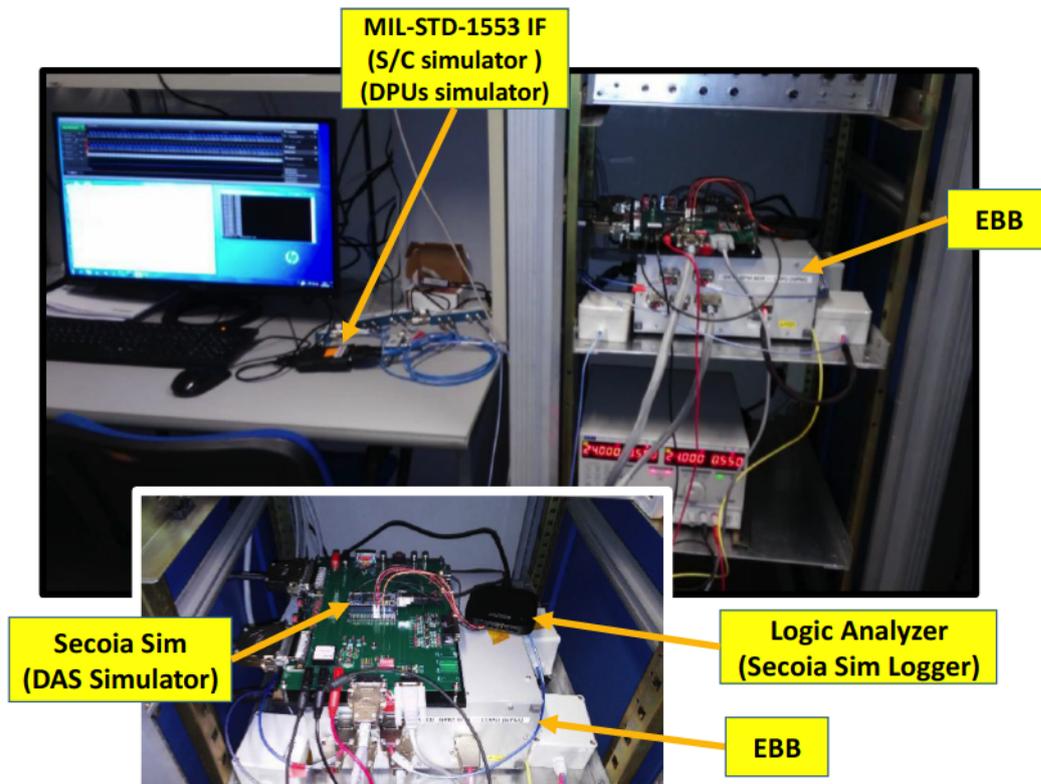


Figure 3.9: Setup of NI-ICU Application Software development platform, composed of the NI-ICU bread board (EBB) and related test equipment.

### 3.2.6 The MIL-STD-1553 interface

MIL-STD-1553 is a data bus military standard. Hardware elements are 3: Bus Controllers, Remote Terminals and Bus Monitors. These terminals can be connected to the bus through 2 methods: *direct coupling* and *transformer coupling* (see Figure 3.10). In the *direct coupling*, the resistors are located within the terminal, while for transformer coupling, they are placed with the coupling transformer in boxes called *data bus couplers*, allowing the stubs for terminal connection to the bus to be longer (609.6 cm vs. 30.48 cm). For the actual setup of our simulators, transformer couplers have been used. The Bus Controller (BC) is responsible for managing the data flow along the bus and is the only terminal allowed to issue commands on the bus, for data transfer or bus management (defined as *mode commands* with their specific *mode codes*). Only one Bus Controller at a time may be active on the line. A Bus Monitor (BM) may be used to collect information about the exchange of (filtered/unfiltered) data on the bus. Finally, the standard defines Remote Terminals (RT) as “all terminals not operating as the Bus Controller or as a Bus Monitor”, so that in general a Remote Terminal includes the electronics in charge of transferring data from the bus to a set of subsystems, i.e. the devices which use the data being transferred. A Remote Terminal is not just a data formatter, it is

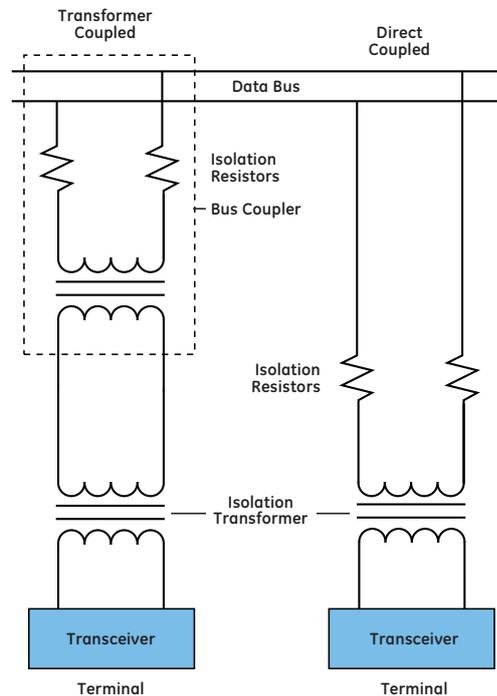


Figure 3.10: Terminal connection configurations for MIL-STD-1553 interface.

a terminal capable of properly responding to the Bus Controller in case of data transmission request according to the standard protocol, buffering messages in specific memory addresses (defined as Sub-Addresses, 32 for each Remote Terminal) and reporting the status of the communication.

### The 1553 protocol

The term “protocol” refers to the rules under which the transfer of data on the bus occurs. Every message being transmitted on the bus is made up by three types of words: *command words*, *data words* and *status words*. The bit map of each of these three word types is shown in Figure 3.11. The *command word* specifies the function that the Remote Terminal has to perform and only the Bus Controller can transmit it; the *data word* represents the actual information carried by the message being transferred (a single message can carry a maximum number of 32 data words); the *status word* contains the information about the status of the communication and of the interested Remote Terminal. The exchange of data on the bus is based on message transmission; the standard defines ten types of message formats usually divided into 2 groups: the “information transfer formats” (Figure 3.13) and the “broadcast information transfer formats” (Figure 3.14). The difference between these 2 groups is that broadcast messages are transmitted simultaneously to multiple Remote Terminals, so that the status word transmission is suppressed in order to avoid a superposition of messages coming

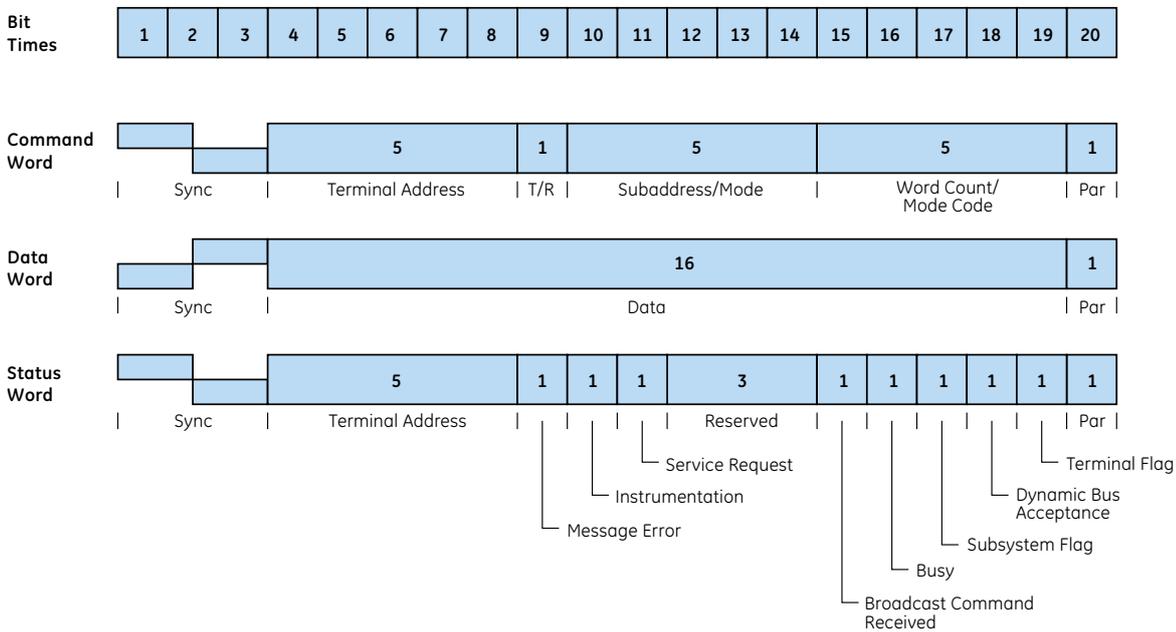


Figure 3.11: *Word formats for the 1553 protocol [43].*

<b>Command Word</b>	20 $\mu$ Sec
<b>Status Word</b>	20 $\mu$ Sec
<b>Data Word(s)</b>	20 $\mu$ Sec (each)
<b>Inter Message Gap</b>	2 $\mu$ Sec minimum
<b>Response Time</b>	10 $\mu$ Sec maximum

Figure 3.12: *Bus loading budget for a MIL-STD-1553 data bus.*

from the terminals. The general principle on which the communication is based, is that after the successful accomplishment of the transmission of a message, the Remote Terminal notifies the event to the Bus Controller by sending a status word, and this is the way through which the data exchange is validated. Our test equipment does not require the communication between Remote Terminals and so RT-RT messages are not used. A message sent from the Bus Controller to a Remote Terminal (BC-RT) is defined as “Receive Command”, whereas a message from a Remote Terminal to the Bus Controller (RT-BC) is called “Transmit Command”. The naming convention has been stated to reflect the perspective of the Remote Terminal, which is respectively receiving and transmitting the message in the 2 cases. Figure 3.12 reports the amount of time the bus remains busy during the transmission of a specific type of data structure (*bus loading budget*).

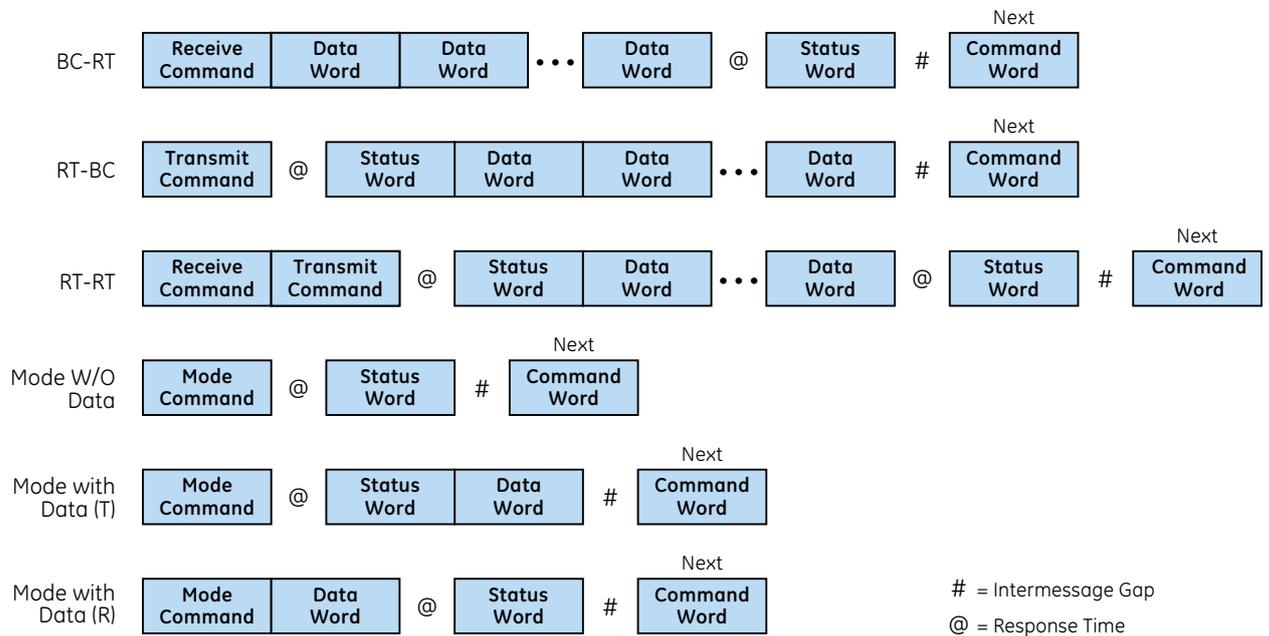


Figure 3.13: Information transfer formats. The response time is the time required for the status word to be received.

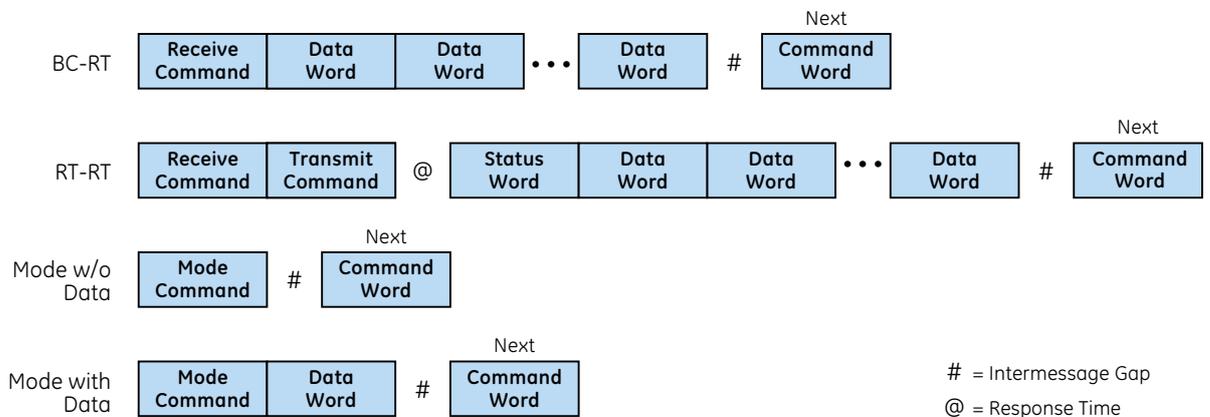


Figure 3.14: Broadcast information transfer formats. No status words are sent back from the Remote Terminals to the Bus Controller.

### 3.2.7 HW/SW configuration for MIL-STD-1553 interface simulators

In order to setup a platform to develop the NI-ICU ASW, the following functionalities shall be provided by MIL-STD-1553 boards:

- Bus Controller (BC) connected to the *science bus* to simulate the communication with

the spacecraft (S/C Simulator);

- Remote Terminals (RTs) connected to the *internal bus* to simulate the communication with NI-DPUs (NI-DPUs Simulator);
- Bus Monitor (BM) for optional monitoring of the bus(es).

The minimal setup to test the NI-ICU ASW requires at least 1 BC and 1 RT; the full setup is realized with 1 BC, 2 RTs and 2 BMs. The selected board is the USB 1553 Avionics interface produced by Astronics Ballard Technology [44]. The Ballard USB 1553 Avionics interface is a portable USB 2.0 compliant device for communicating with MIL-STD-1553 systems. The design of this test equipment is flexible enough to manage several possible configurations such as with/without bus monitoring or with/without error injection. A working setup can be achieved in different ways depending on the Ballard models available. Some examples are listed below:

- Ballard UA1133 (2 channels, multi-function) can be configured to provide all the functionalities described above (BC, 1 or 2 RTs, 2 BMs) and have also protocol error-injection capability;
- Ballard UA1122 (2 channels, single-function) can be configured as BC and 1 or 2 RTs;
- 2 Ballard UA1120 (1 channel, single-function) devices can be configured respectively as BC and 1 or 2 RTs.

The science and internal bus networks are realized with accessories such as bus cables, transformer couplers and bus terminators compliant for use with MIL-STD-1553 avionics databus interfaces. The connection between the bus stub (PL75 connector) and the EBB (DB9 connector) is done with stub cable adapters which host also further transformer couplers. The S/C and NI-DPUs simulator softwares are developed using the C++11 programming language and the BOOST libraries (<https://www.boost.org>). The build process is performed using the cross-platform and compiler-independent CMAKE tool. The software is written with the purpose to be compiled both on Windows and Linux operating systems. The source code is maintained in a Git repository (<https://baltig.infn.it/euclid/ni-iss-devel>), with a Continuous Integration capability.

### 3.3 S/C simulator SW design

The S/C simulator software design is based on several layers, as shown in Figure 3.15. Each layer is identified by a namespace and implements specific functionalities. The Ballard's APIs are wrapped in a set of classes which embed direct calls to the board. The list of namespace is (starting from the bottom of Figure 3.15):

- *bti*, which wraps the Ballard's APIs into a set of C++ classes;
- *mil1553*, which includes some classes to format 1553 messages;
- *pus* (at the Bus Protocol level) includes classes to manage messages for the science bus protocol, which implements some PUS services;

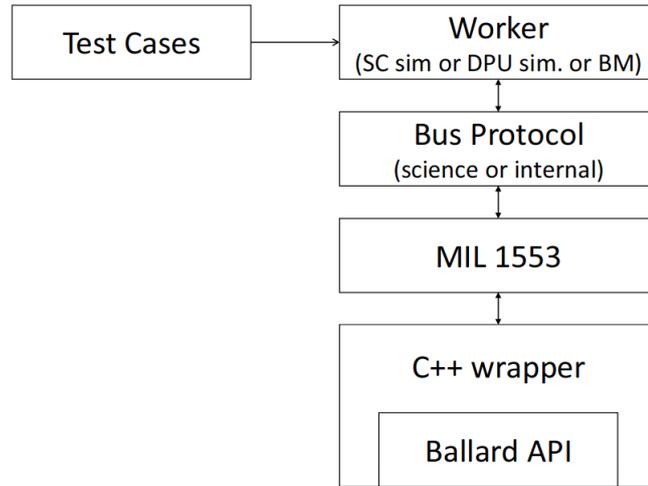


Figure 3.15: *S/C simulator software conceptual structure.*

- *task* (at the Worker level), which implements the main worker classes (S/C simulator, NI-DPU simulator, Bus Monitor);
- *test\_cases*, which is a series of classes that read the TC/TM test sequence and send them to the worker with the proper timing.

The Ballard USB device is accessed through a Card object. The Card initializes a Core, which, in turn, enables the communication on the channel(s). Each channel shall be configured as BC, RTs and/or BM. For example, in case of a 2-channel multi-functional device, a BC is created on channel 0 to simulate the S/C; 1 or 2 RTs are created on channel 1 to simulate the DPU(s); a BM is created on channel 0 to verify the communication; BM is set on channel 1 as well. Then, the worker objects (S/C simulator, NI-DPU simulator and/or BMs) are created. Each worker has a method which implements the communication protocol. In case of S/C simulator, it schedules a bus profile according to the Spacecraft Communication Interface Control Document [33]. Each worker logs relevant information (time, message name, message data, etc.) using `boost::log`. The program stops when the queue of messages to be transmitted is ended or on user request. The software is designed to use events from the Ballard devices: these events are managed by the Boost asynchronous input/output library. This allows the execution of the application with only one thread, avoiding possible conflicts due to concurrent access to the MIL-STD-1553 resources.

The S/C simulator implements the communication protocol with bus profile and PUS services as defined in the Spacecraft Communication Interface Control Document. The science bus profile is reported in Figure 3.17. Figure 3.18 shows the time scheduling of messages sent during a so-called Communication Frame (1/60 seconds). During the initialization phase, a data feeder routine (implemented as a Sequencer object) reads a sequence of telecommands (TCs) from a file and stores them in memory. The transmission of the first TC message starts as soon as the reset procedure has been successfully accomplished (according to the MIL-STD-1553 System

Data Bus Protocol Specification Document [45]). The Reset is performed as follows:

- the S/C simulator sends a DTD (Distributor Transfer Descriptor, Figure 3.19) message with the reset bit ON to the EBB;
- once received, the EBB sends back a corresponding DTC (Distribution Transfer Confirmation, Figure 3.20) with the reset bit OFF.

The exchange of TM packets is realized through the procedure reported in Figure 3.26, using the Acquisition Transfer Request (ATR, Figure 3.24) and the Acquisition Transfer Confirmation (ATC, Figure 3.25) messages from ECSS acknowledgement protocol. The TCs are converted to PUS format packets and transmitted into the next available slot according to the bus profile. Figure 3.21 shows the PUS format implemented for TCs, while Figure 3.22 illustrates the one for TM packets. For example, a test file can contain the sequence to initialize the NISP nominal observation mode [46]:

- Load the Dither Configuration Table (DITH\_CONFIG\_TAB);
- Wait (10 seconds);
- Load the Processing Parameters Table (PROC\_PARAM\_TAB);
- Wait (10 seconds);
- Load the SCE (Sensor Chip Electronics, i.e. SIDECAR) Exposure Table (SCE\_EXP\_TAB).

The corresponding output log file is shown in Figure 3.16, where one can see the transmitted TCs, the ECSS protocol messages and the received TMs (PUS 1,1/1,2 and PUS 1,7/1,8) data words. The procedure for a telecommand block exchange is described in Figure 3.23.

### 3.3.1 S/C simulator communication test

A test was conducted to check NI-ICU and NI-DPU electrical connections and communication operation via MIL-STD-1553 buses. The test was performed using the NI-ICU EBB prototype and the Maxwell SCS 750 P board, which is part of the NI-DPU DM (Demonstration Model). The main purpose of the test was to verify the two prototype boards compatibility via MIL-STD-1553 links. Some basic 1553 drivers features were tested and some simple TCs were executed, thus implying the presence of a S/C interface simulator.

#### Test bench

The hardware configuration for the test is schematically shown in Figure 3.27. Although the configuration is not representative of the NI-ICU/NI-DPU complete setup, it was sufficient to test the MIL-STD-1553 communication interfaces between the devices. A Maxwell SCS 750 P single board computer was used as NI-DPU simulator. Figure 3.28 illustrates how the Maxwell board was mounted in a commercial rack and equipped with a MIL-STD-1553 bus. The hardware architecture of the Maxwell board is shown in Figure 3.29, and the Board Support Package used for the test was the one released by the Maxwell Company. For NI-

```

bc_service.log (14.8 KB) - Notepad++
File Edit View Preferences Help
C:\Users\Yto\Desktop\log\bc_service.log (14.8 KB)
bc_service.log ch1-monitor.log main.log
[11:11:12.472]: new operation as a TC/TM
[11:11:12.472]: BC -> TC_00 : 0400 0001 0014 1908 0100 000c 4014 4814 3101 0102 0000 000f 0010 4808 0000 0000 0001 dff8*
[11:11:12.482]: 2382744 BC -> DTD : 0024 2a51
[11:11:12.722]: 2602449 BC -> ATR2 : 001a 2a01
[11:11:12.762]: 2720424 BC -> TM2_00 : 0400 c001 0013 1003 1901 0000 0000 0000 0003 0000 0000 0000 7b44
[11:11:12.772]: 2732704 BC -> AIC2 : 001a 2a01
[11:11:13.462]: 3431783 BC -> DTC : 0024 2a01
[11:11:13.712]: 3681503 BC -> ATR2 : 001a 2a02
[11:11:13.762]: 3719484 BC -> TM2_00 : 0400 c002 0013 1003 1901 0000 0000 0000 0003 0000 0000 0000 15e8*
[11:11:13.772]: 3731764 BC -> AIC2 : 001a 2a02
[11:11:14.714]: 4680803 BC -> ATR2 : 0016 2a03
[11:11:14.764]: 4718844 BC -> TM2_00 : 0400 c003 000f 1001 0101 0000 0000 0000 1400 c001 a8fa
[11:11:14.764]: 4730824 BC -> AIC2 : 0016 2a03
[11:11:15.718]: 5679623 BC -> ATR2 : 0016 2a04
[11:11:15.768]: 5737604 BC -> TM2_00 : 0400 c004 000f 1001 0701 0000 0000 0000 1400 c001 dff6
[11:11:15.768]: 5737604 BC -> AIC2 : 0016 2a04
[11:11:16.708]: 6678683 BC -> ATR2 : 001a 2a05
[11:11:16.758]: 6716664 BC -> TM2_00 : 0400 c005 0013 1003 1901 0000 0000 0000 0003 0000 0000 0000 1ca4
[11:11:16.768]: 6728944 BC -> AIC2 : 001a 2a05
[11:11:17.708]: 7679743 BC -> ATR2 : 001a 2a06
[11:11:17.758]: 7718724 BC -> TM2_00 : 0400 c006 0013 1003 1901 0000 0000 0000 0003 0000 0000 0000 7254
[11:11:17.768]: 7728004 BC -> AIC2 : 001a 2a06
[11:11:18.468]: 8369984 BC -> TC_00 : 0400 c007 0017 1908 0100 000e 40e4 0400 0104 0210 0410 0410 0310 0410 0000
[11:11:18.708]: 8676803 BC -> ATR2 : 001a 2a07
[11:11:18.758]: 8714784 BC -> TM2_00 : 0400 c007 0013 1003 1901 0000 0000 0000 0003 0000 0000 0000 a7a2
[11:11:18.758]: 8727064 BC -> AIC2 : 001a 2a07
[11:11:19.488]: 9428143 BC -> DTC : 0016 2a02
[11:11:19.708]: 9679843 BC -> ATR2 : 001a 2a08
[11:11:19.758]: 9713844 BC -> TM2_00 : 0400 c008 0013 1003 1901 0000 0000 0000 0003 0000 0000 0000 6184
[11:11:19.758]: 9726124 BC -> AIC2 : 001a 2a08
[11:11:20.708]: 10474923 BC -> ATR2 : 001a 2a09
[11:11:20.758]: 10714904 BC -> TM2_00 : 0400 c009 0013 1003 1901 0000 0001 0001 0001 0003 0000 0000 0000 b4a2
[11:11:20.758]: 10725104 BC -> AIC2 : 001a 2a09
[11:11:21.708]: 11673983 BC -> ATR2 : 001a 2a0a
[11:11:21.758]: 11731964 BC -> TM2_00 : 0400 c00a 0013 1003 1901 0000 0001 0001 0001 0003 0000 0000 0000 da98
[11:11:21.758]: 11724244 BC -> AIC2 : 001a 2a0a
[11:11:22.708]: 12473043 BC -> ATR2 : 0016 2a08
[11:11:22.758]: 12711024 BC -> TM2_00 : 0400 c00b 000f 1001 0101 0000 0001 0001 1400 c002 1185
[11:11:22.758]: 12723304 BC -> AIC2 : 0016 2a0b
[11:11:23.708]: 13672104 BC -> ATR2 : 0016 2a0c
[11:11:23.758]: 13710084 BC -> TM2_00 : 0400 c00c 000f 1001 0701 0000 0001 0001 1400 c002 8a18
[11:11:23.758]: 13722364 BC -> AIC2 : 0016 2a0c
[11:11:24.468]: 8364344 BC -> TC_00 : 0400 0014 003d 1908 0100 000e 40c7 001b 030c 0000 0000 0000 0000 0000 0000 0000
[11:11:24.468]: 0 BC -> TC_01 : 0008 9a84
[11:11:24.468]: 8376184 BC -> DTD : 0044 2a03
[11:11:24.708]: 14671163 BC -> ATR2 : 001a 2a0d
[11:11:24.758]: 14709144 BC -> TM2_00 : 0400 c00d 0013 1003 1901 0000 0001 0001 0001 0003 0000 0000 0000 d319
[11:11:24.758]: 14721424 BC -> AIC2 : 001a 2a0d

```

Figure 3.16: S/C simulator software log file.

ICU, the hardware setup was the one described in §3.2.5 and shown in Figure 3.9. On a first desktop PC, GRMON [41] and Eclipse IDE (<http://www.eclipse.org/downloads/packages/eclipse-ide-cc-developers/oxygen1a>) softwares were running, in order to (respectively) upload applications on EBB FPGA via the EBB Test Interface and run them in debug mode. A second desktop PC with a Ballard 1553 USB interface was used for the S/C simulator (or for the running of CoPilot, an advanced testing tool for Ballard USB devices). The Bus Monitors were implemented:

- on the PC running the Maxwell board application software (using CoPilot Databus Tool version 7.00 with a Ballard UA1122 device) for the NI-DPU/NI-ICU interface;
- on the PC running the S/C interface simulator (using the S/C simulator software with a Ballard UA1133 device) for the NI-ICU/spacecraft interface.

### Application softwares

For the NI-DPU, a custom test software and the current version of the NI-DPU ASW were used during the test. The NI-DPU ASW was running on *VxWorks 5.1* [47] real-time operating system, while the custom test software consisted in an application running as Remote Terminal (RT) on a MIL-STD-1553 communication protocol. The DPU-ASW was implemented by the

CF	MMU Nom					MMU Red					VIS					NISP				
	send TC	send DTD	poll ATR	get TM	send ATC	send TC	send DTD	poll ATR	get TM	send ATC	send TC	send DTD	poll ATR	get TM	send ATC	send TC	send DTD	poll ATR	get TM	send ATC
0			1																	
1	1	1																		
2				1	1															
3	1	1	1																	
4										1	1	1								
5				1	1															
6	1	1	1																	
7												1	1							
8				1	1															
9	1	1	1																	
10															1	1	1			
11				1	1															
12	1	1	1																	
13																		1	1	
14				1	1															
15	1	1	1																	
16						S	S	S												
17				1	1															
18	1	1	1																	
19								S	S											
20				1	1															
21	1	1	1																	
22												1								
23				1	1															
24	1	1	1																	
25													1	1						
26				1	1															
27	1	1	1																	
28																		1		
29				1	1															
30	1	1	1																	
31																			1	1
32				1	1															
33	1	1	1																	
34						S	S	S												
35				1	1															
36	1	1	1																	
37								S	S											
38				1	1															
39	1	1	1																	
40										S	S	S								
41				1	1															
42	1	1	1																	
43												S	S							
44				1	1															
45	1	1	1																	
46															S	S	S			
47				1	1															
48	1	1	1																	
49																		S	S	
50				1	1															
51	1	1	1																	
52						S	S	S												
53				1	1															
54	1	1	1																	
55								S	S											
56				1	1															
57	1	1																		
58																				
59	1	1																		
<b>TM/TC Summary</b>	21			19		3S			3S		1+1S			2+1S		1+1S			2+1S	

Figure 3.17: Science bus scheduling profile: the major time frame of 1 second is subdivided in communication frames (CF) of 1/60 seconds. NISP communicates with the S/C only during CF highlighted in green (1 is for nominal allocation, S for “spare”). ATR and ATC, DTD and DTC are respectively ECSS acknowledgement protocol messages for TM and TC exchange [33] (they are described in §3.3).

Relative time wrt CF start ALLOCATION [ $\mu$ s]	Slot allocated duration [ $\mu$ s]	Message type	Remark
0	100	CF synchronisation message	Broadcast message. For CF=0 synch without DW (msg duration is 32 microsec) For CF=1-59 synch with DW (msg duration is 52 microsec)
100	200	time message	Broadcast message. Only in CF=33
300	725	Data transfer block (TM/TC) #1	Most significant message
1025	725	Data transfer block (TM/TC) #2	
1750	725	Data transfer block (TM/TC) #3	
2475	725	Data transfer block (TM/TC) #4	
3200	725	Data transfer block (TM/TC) #5	
3925	725	Data transfer block (TM/TC) #6	
4650	725	Data transfer block (TM/TC) #7	
5375	725	Data transfer block (TM/TC) #8	
6100	725	Data transfer block (TM/TC) #9	
6825	725	Data transfer block (TM/TC) #10	
7550	725	Data transfer block (TM/TC) #11	
8275	725	Data transfer block (TM/TC) #12	
9000	725	Data transfer block (TM/TC) #13	
9725	725	Data transfer block (TM/TC) #14	
10450	725	Data transfer block (TM/TC) #15	
11175	725	Data transfer block (TM/TC) #16	
11900	150	Packet Control #1	BC polls DTC (in case of reset only)
12050	150	Packet Control #2	BC sends DTD
12200	150	Packet Control #3	BC polls ATR
12350	150	Packet Control #4	BC sends ATC
12500	725	Units Wrap-Around	Command W/A or acquire W/A data (S/A=30R/T)
13225	725	Health monitoring data (S/A=1T) TM Block Retrieval Reset Request (S/A=1R) Mode Commands (S/A=31R)	The health monitoring data from S/A=1T is acquired cyclically every second for each enabled MMU RT: - MMU A (RT=1) acquired on CF=0 - MMU B (RT=2) acquired on CF=1 TM Block Retrieval Reset Request (S/A=1R) and Mode Commands (S/A=31R) are sent on-request.
13950	1450	Spare	2 slots
15400	1266	Idle	
16666			

Figure 3.18: Time scheduling of messages sent during a single Communication Frame [33].

1. Data word (Distribution Data size)		2. Data word (Distribution Control)				
( 4 Bit)	(12 Bit)	( 1 Bit)	( 1 Bit)	( 1 Bit)	( 5 Bit)	( 8 Bit)
Reserved '0000'	Number of Bytes for the current data block (SIZE)	QoS '0'	Reset	Mode '1'	Receive TC Block SA	Distribution Block Count (DBC)

Note: QoS = '0' means 'best effort' and Mode = 1 means 'deep' sub-addressing mode.

Figure 3.19: Distribution Transfer Descriptor (DTD) data words.

1. Data word (Distribution Data size)		2. Data word (Distribution Control)				
( 4 Bit)	(12 Bit)	( 1 Bit)	( 1 Bit)	( 1 Bit)	( 5 Bit)	( 8 Bit)
Reserved '0000'	Number of bytes for the current data block (SIZE)	Error '0'	Reset	Mode '1'	Receive TC Block SA	Last received Distribution Block Count (DBC)

Note: QoS = '0' means 'best effort' and Mode = 1 means 'deep' sub-addressing mode.

Figure 3.20: Distribution Transfer Confirmation (DTC) data words.

Packet Header						Packet Data Field			
Packet ID				Packet Sequence control		Packet data length	Data Field Header	Application data	Packet Error Control
Version number	Packet Type	Data Field header Flag	APID	Flags	Count				
3 bits	1 bit	1 bit	11 bits	2 bits	14 bits				

**TC Packet Format**

The data field header is as follows:

CCSDS Secondary Header Flag	TC Packet PUS Version number	Ack	Service Type	Service Subtype	Source ID
1 bit	3 bits	4 bits	1 Byte	1 Byte	1 Byte

Figure 3.21: PUS format for TC packets implemented in NISP.

Packet Header						Packet Data Field			
Packet ID				Packet Sequence control		Packet data length	Data Field Header	Source data	Packet Error Control
Version	Type	Data field header Flag	APID	Grouping Flags	Sequence Count				
3 bits	1 bit	1 bit	11 bits	2 bits	14 bits				
16 bits				16 bits		16 bits	10 Bytes	N*16 bits (max 1006 Bytes)	16 bits

TM Packet Format

The data field header is composed as follows:

Spare	TM Packet PUS Version number	Spare	Packet type	Packet Subtype	Destination ID	Time
1 bit	3 bits	4 bits	1 Byte	1 Byte	1 Byte	6 Bytes

Figure 3.22: PUS format for TM packets implemented in NISP.

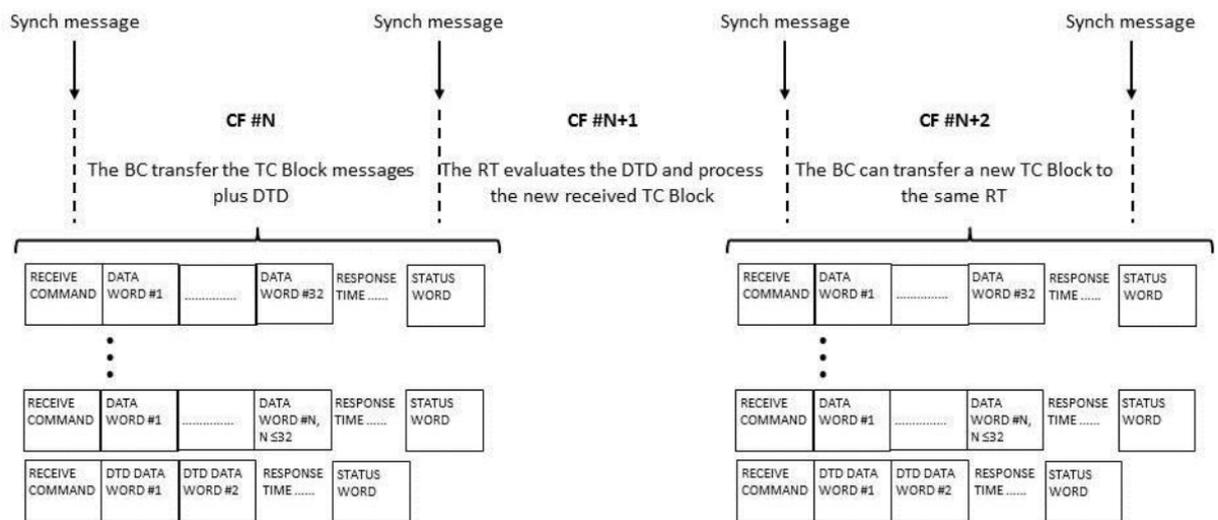


Figure 3.23: Schematic representation of the telecommand block transfer procedure.

1. Data word (Acquisition Data size)		2. Data word (Acquisition Control)				
( 4 Bit)	(12 Bit)	( 1 Bit)	( 1 Bit)	( 1 Bit)	( 5 Bit)	( 8 Bit)
Reserved '0000'	Number of bytes for the current data block (SIZE)	QoS '0'	Reset	Mode '1'	Transmit TM Block SA	Acquisition Block Count (ABC)

Note: QoS = '0' means 'best effort' and Mode = 1 means 'deep' sub-addressing mode.

Figure 3.24: Acquisition Transfer Request (ATR) data words.

1. Data word (Distribution Data size)		2. Data word (Distribution Control)				
( 4 Bit)	(12 Bit)	( 1 Bit)	( 1 Bit)	( 1 Bit)	( 5 Bit)	( 8 Bit)
Reserved '0000'	Number of bytes for the current data block (SIZE)	Error '0'	Reset	Mode '1'	Transmit TM Block SA	Acquisition Block Count (ABC)

Note: QoS = '0' means 'best effort' and Mode = 1 means 'deep' sub-addressing mode.

Figure 3.25: Acquisition Transfer Confirmation (ATC) data words.

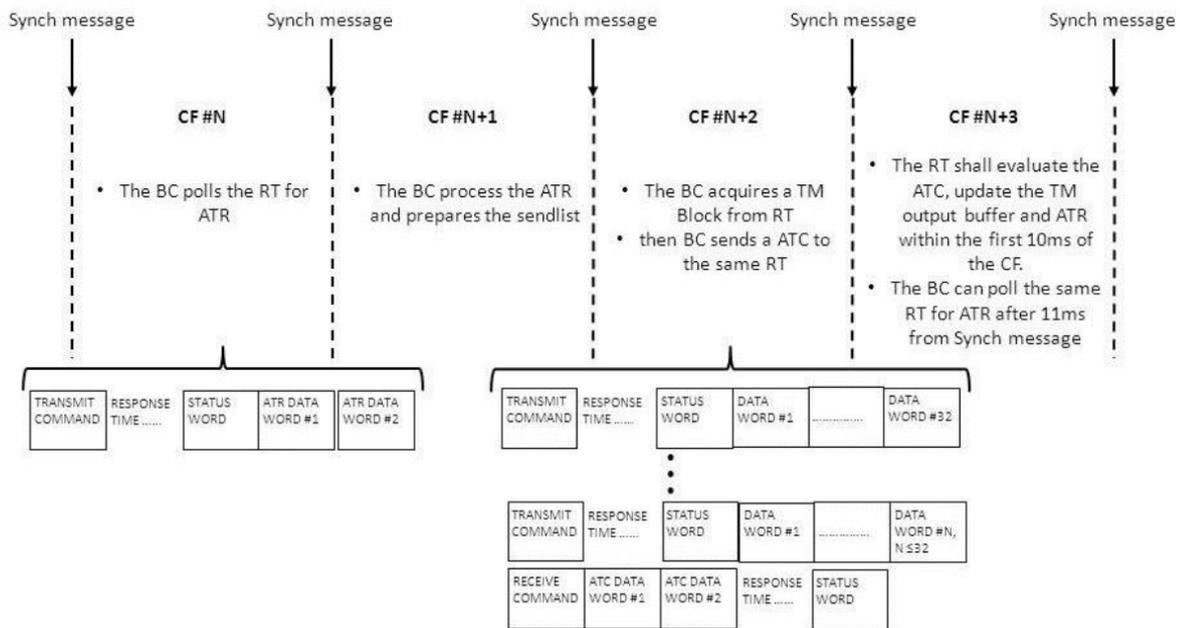


Figure 3.26: Schematic representation of the telemetry block transfer procedure.

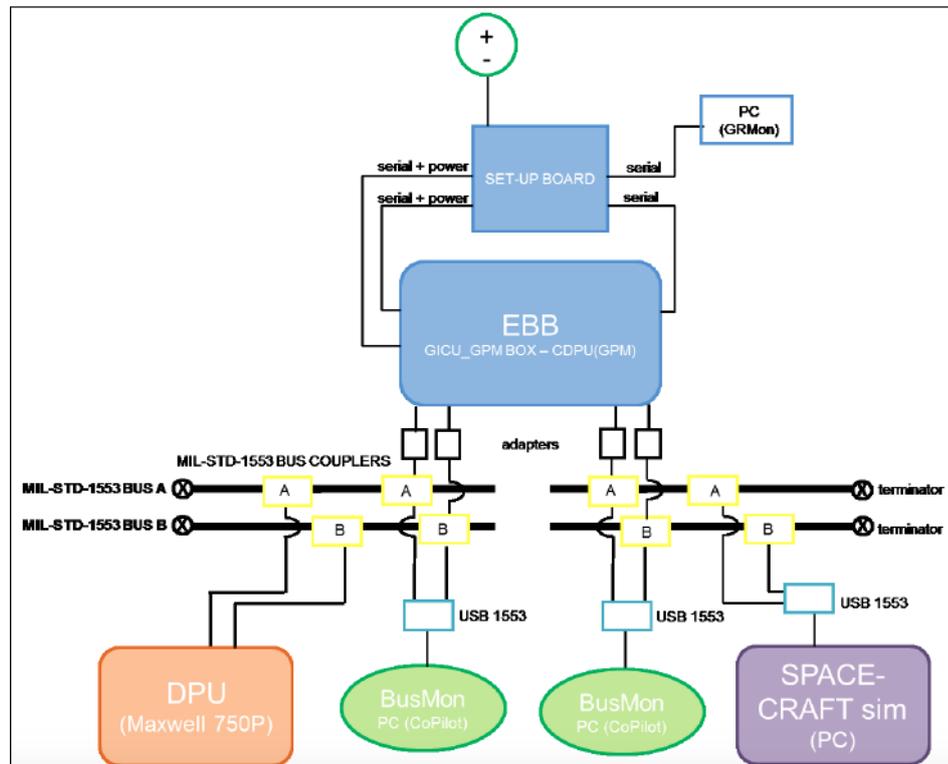


Figure 3.27: Test setup diagram: note that in the MIL-STD-1553 connection between ICU-EBB and DPU + Bus Monitor PC, only bus B was used.

use of the VxWorks interrupt technique, based on immediate reaction to external events (represented in this case by any kind of activity occurring on the MIL-STD-1553 bus). This technique consists in a standard *hardware interrupt handling*, which is the usual mechanism adopted to provide a system with information concerning external events running in a specific context. The set of routines managing the interrupt handling was commanded by the use of a simple ASCII interface, so that the NI-DPU ASW was able to handle the following telecommands on dedicated Sub-Addresses:

- DITH\_CONFIG\_TAB (*receive mode*);
- PROC\_PARAM\_TAB (*receive mode*);
- SCE\_EXP\_TAB (*receive mode*);
- ASW\_STATUS\_TAB (*transmit mode*).

The arrival of a SCE\_EXP\_TAB command triggers the automatic start of data processing (whose evaluation is not part of the test goal). Once the processing ends, an ASW\_STATUS\_TAB command is prepared and made available to the NI-ICU Bus Controller (BC) on the MIL-STD-1553 bus at a dedicated Sub-Address. Figure 3.30 schematizes a block diagram of the NI-DPU

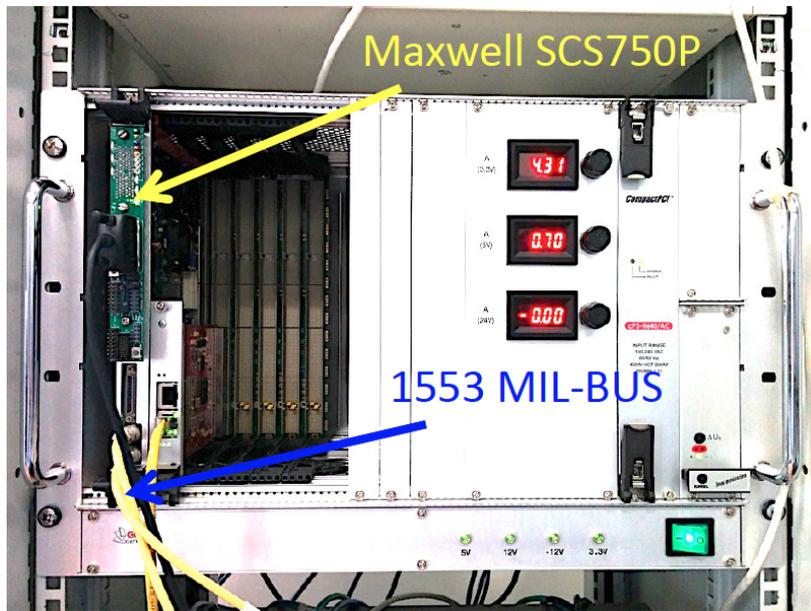


Figure 3.28: Maxwell SCS 750 P board mounted in commercial rack; MIL-STD-1553 buses A, B are attached to the yellow cables.

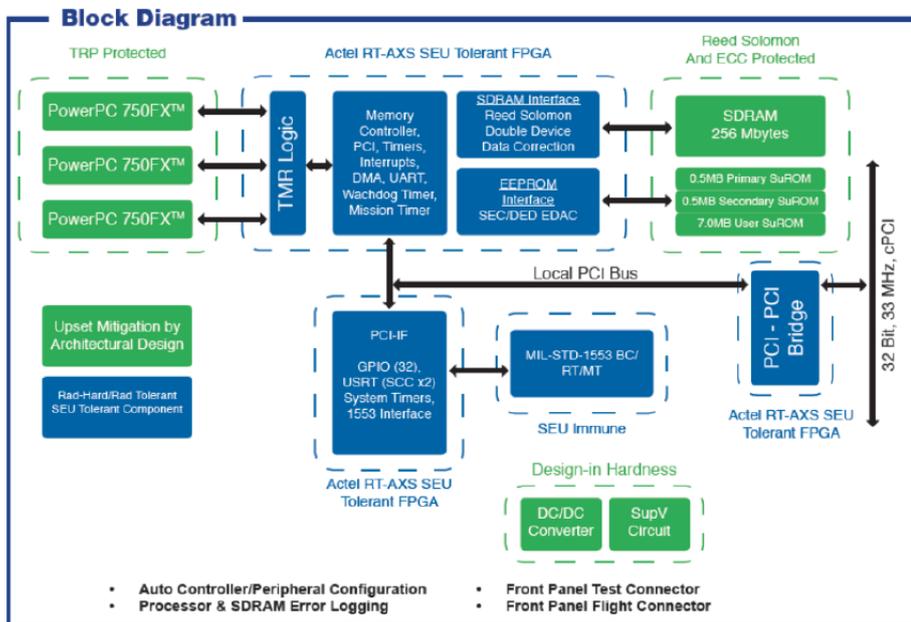


Figure 3.29: Block diagram of the hardware configuration of the Maxwell SCS 750 P board.

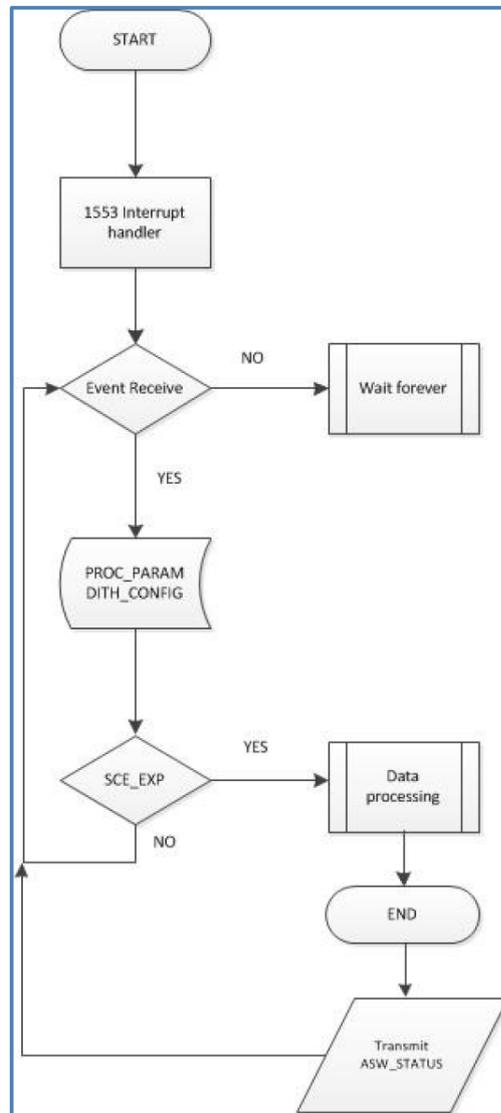


Figure 3.30: Block diagram showing the operational flow of the NI-DPU ASW.

ASW.

The NI-ICU ASW prototype comprised two stand-alone test programs: the first implementing a real-time algorithm acting as a RT and managing TC/TM exchange with the S/C, while the second acting as a BC in order to exchange commands and configuration tables with the NI-DPU. The NI-ICU ASW structure has been largely described in §3.2, here it is recalled that it provides:

- the scheduling of the communication with the S/C and with the NI-DPU;

- the configuration and management of the Sub-Addresses (at MIL-STD-1553 level);
- towards the S/C: the management of the *Acquisition Transfer Request* (ATR) and *Acquisition Transfer Confirmation* (ATC) messages for TM exchange; the management of the *Distribution Transfer Descriptor* (DTD) and *Distribution Transfer Confirmation* (DTC) messages for TC transmission;
- towards the NI-DPU: the dispatching of commands and tables received from the S/C via TCs, the decoding and retrieving of asynchronous tables received from the NI-DPU itself.

For the purpose of this test, only PUS services 1, 3 and 8 were implemented by the NI-ICU ASW, the others being unnecessary.

Finally, concerning the S/C simulator software, the version used for the test was provided with a sequencer routine, which read data from a test sequence file and stored them in the stack. The test sequence file contained the NISP observation sequence and TCs data interleaved with 5 seconds gaps. Once TCs data are stored in the local memory, they are converted to PUS format packets and sent to NI-ICU through Ballard 1553 Driver's API. The S/C simulator software acts as a BC and manages ATR/ATC, DTD/DTC and the scheduling of messages through Ballard USB Card's API. Every time a DTC is received from NI-ICU ASW, a comparison with the latest DTD sent is performed, and if they are found equal, the next DTD counter field is incremented. In the same way, each time a new ATR message is received from NI-ICU ASW, a new TM packet is read at the specified Sub-Address and an ATC message (containing last ATR information) is sent back to NI-ICU.

Both the S/C simulator and NI-ICU ASW versions used for the test were properly tagged and made available at the previously mentioned Git server (<https://baltig.infn.it>).

## Test plan

The test plan consisted of 3 steps; each of them was mandatory to perform the next one:

1. *single message exchange in receive (R) and transmit (T) mode via MIL-STD 1553*: NI-ICU EBB acts as BC, Maxwell board as RT and CoPilot software as BM (Bus Monitor); during this test, the NI-ICU ASW continuously dispatches a set of commands to the NI-DPU, whose ASW is alternatively stopped and re-activated in debug mode for each received message; the Terminal Addresses were fixed in this way: NI-DPU was assigned 14, NI-ICU was assigned 3, broadcast messages (messages sent to all enabled Terminal Addresses) were assigned 31; Table 3.2 shows all the 1553 messages sent (mode code 1 is “synchronize without data word”, mode code 2 is “transmit status word”, mode code 17 is “synchronize with data word” and mode code 21 is “override selected transmitter shutdown”);
2. *exchange of a typical sequence of messages composing an observation cycle*: NI-ICU ASW sends to NI-DPU the messages needed to configure and start an acquisition with no time schedule applied; NI-ICU ASW was stopped and re-activated in debug mode for each message; the Terminal Addresses were fixed as before: NI-DPU was assigned 14, NI-ICU was assigned 3, broadcast messages (messages sent to all enabled Terminal Addresses)

Message #	RT	SA	R/T	Data Word	Success	Comment
1	14	3	R	Yes	No	Maxwell driver limitation, failure on the first message: Maxwell board does not process the first 1553 received message
2	14	3	R	Yes	Yes	
3	31	3	R	Yes	No	Maxwell driver limitation in changing RT, broadcast limitation
4	14	3	T	Yes	Yes	
5	31	0	T	Mode code 1	No	Maxwell driver limitation in changing RT, broadcast limitation
6	31	3	T	Mode code 1	No	Maxwell driver limitation in changing RT, broadcast limitation
7	14	0	T	Mode code 1	Yes	
8	14	31	T	Mode code 1	Yes	
9	14	0	R	Mode code 17	No	NI-ICU driver limitation, no reception of status word
10	14	31	R	Mode code 17	No	NI-ICU driver limitation, no reception of status word
11	14	0	T	Mode code 2	Yes	
12	14	0	R	Mode code 21	No	NI-ICU driver limitation, no reception of status word

Table 3.2: Table summarizing the results of the first step of the test.

were assigned 31; Table 3.3 shows the exchanged commands; as a second trial, exposure commands (hardcoded into NI-ICU ASW) were exchanged within a schedule with 20 seconds gap between each message (1 to 4) and also this test was successful;

Message #	RT	SA	R/T	Data Word	Success
1	14	19	R	PROC_PARAM_TAB	Yes
2	14	21	R	DITH_CONFIG_TAB	Yes
3	14	18	R	SCE_EXP_TAB	Yes
4	14	18	T	ASW_STATUS_TAB	Yes

Table 3.3: Table summarizing the results of the second step of the test.

3. *integration of NI-DPU and NI-ICU ASWs with S/C simulator*: the final step was conceived to test the following flow:

- S/C simulator sends the TCs to command an observation sequence with the corresponding DTDs: PROC\_PARAM\_TAB, DITH\_CONFIG\_TAB and SCE\_EXP\_TAB;

- NI-ICU ASW receives a DTD and prepares the corresponding DTC providing PUS TM(1,1)/TM(1,2) for the acceptance success/failure of the TC and PUS TM(1,7)/TM(1,8) for the execution status report;
- NI-ICU ASW decodes the TC and transmits it to the NI-DPU;
- NI-DPU ASW receives the command and starts the acquisition;
- NI-DPU ASW sends back the ASW\_STATUS\_TAB to the NI-ICU on request;
- NI-ICU ASW reads the ASW\_STATUS\_TAB every second;
- NI-ICU ASW, starting from parameter values included in the ASW\_STATUS\_TAB, generates a PUS TM(3,25) packet (NI-DPU HK);
- NI-ICU ASW sends the NI-DPU HK packet to S/C simulator with the corresponding ATR;
- S/C simulator receives the ATR, replies generating and sending back the correct ATC and finally reads the TM packet;

a couple of errors were observed during this last test step by the NI-ICU ASW, concerning:

- the decodification of the DITH\_CONFIG\_TAB received from S/C simulator;
- the formatting of the DPU HK PUS packet;

anyway, these errors were simply overcome in the following way:

- NI-ICU ASW was temporarily patched to transmit the DITH\_CONFIG\_TAB using hardcoded data;
- the Bus Monitor confirmed that DPU HK packet data transmitted on the bus were correct, so an error was present in the routine generating the packet;

after the end of the test, both errors were properly corrected; by the way, the application of the aforementioned modifications allowed to successfully conclude the test.

### **Test final report**

The principal goals of the test were successfully accomplished, namely:

- the MIL-STD-1553 communication between NI-ICU and NI-DPU was properly validated and the Ballard hardware was found suitable to support the software development of the communication interfaces between the NISP Warm Electronics components and the S/C;
- all the telecommands, the data words and the status words (at MIL-STD-1553 protocol level) were correctly transmitted and recognized;
- the S/C interface simulator showed to work quite well and to provide a valid contribution to the test;
- although the electrical connections equipment and configuration were not fully representative of the expected flight arrangement, the test allowed a preliminary integration between NISP subsystems to be verified;

- the first sequence of tests indicated that Maxwell board 1553 drivers present some issues; in particular, failure on the reception of the first message and failure in RT address modification;
- concerning the synchronization between NI-ICU and NI-DPU communication, a more stable and complete version of 1553 drivers and a more precise timing profile were needed for both the devices.

After this first test, other communication tests were performed in order to assess the proper functioning of the ICU-DPU interface as long as their Application Softwares were developed. These tests proved that the communication between the devices is stable and compliant with required message timing and scheduling.

In order to ensure infrared measurements with the NISP instrument, sensors that provide the required performances in noise and quantum efficiency must be used. Here we will discuss infrared detection methods and in particular semiconductor detectors. We will describe the H2RG detectors selected for the Euclid mission, produced by Teledyne, which are able to offer the performances required for the scientific objectives of the instrument. Then we will present a description of the different effects specific to the physics of the H2RG detectors that it is necessary to measure.

## 4.1 Near-Infrared detection principles

Here, we describe the general detection methods which are used for the detection of infrared photons by addressing in more detail the technology of semiconductor detectors. An infrared detection chain converts incident infrared photons into an analog signal (the amplitude of which is proportional to the incident flux) and then into a digital signal in order to record and process the data thus obtained. In order to convert a flux of infrared photons into an electrical signal, two types of detectors can be used:

- *thermosensors*, which detect a signal by a variation of one of their electrical properties (variation of current for thermocouples, or electrical resistance for bolometers) caused by the temperature variation generated by the absorption of infrared photons;
- *photodetectors*, which detect a signal by absorption of the infrared photon and photo-generation of charge carriers (by photoelectric effect); among these, there are semiconductor detectors made of Mercury-Cadmium Telluride (HgCdTe) PN photodiodes, the type of technology used for Euclid's infrared detectors.

In this section, we will discuss the principles of detection of semiconductor photodetectors which allow the conversion of a flux of infrared photons into electrical charges in a semiconductor

crystal. First we will study the concept of a unitary detector and we will then explain the process used to obtain a two-dimensional image. Then, the electronics necessary for reading this signal will be described. Finally, the application of this technology for the creation of a matrix of millions of pixels will be exposed. In current detection systems based on semiconductors, the conversion of a photon flux into a digital signal follows the functional diagram shown in Figure 4.1. An electron can pass from the *valence band* (bound to the atom) to the *conduction band* (bound to the crystal) if it is provided with at least enough energy to cross the *band gap* separating these two bands. It is therefore necessary to provide a quantum of energy  $h\nu$  such that:

$$h\nu = \frac{hc}{\lambda} > \epsilon_g \quad (4.1)$$

with  $\epsilon_g$  the *band gap* in eV,  $h$  the Planck constant ( $6.63 \times 10^{-34}$  J·s),  $\lambda$  the wavelength of the photon in meters and  $c$  ( $3.0 \times 10^8$  m/s) the speed of light. This minimum amount of energy  $\epsilon_g$  allows to define the minimum energy of detectable photons, and therefore to define what is called the *cut-off wavelength*:  $\lambda_c = \frac{hc}{\epsilon_g}$ . There is a wide range of crystals sensitive to a

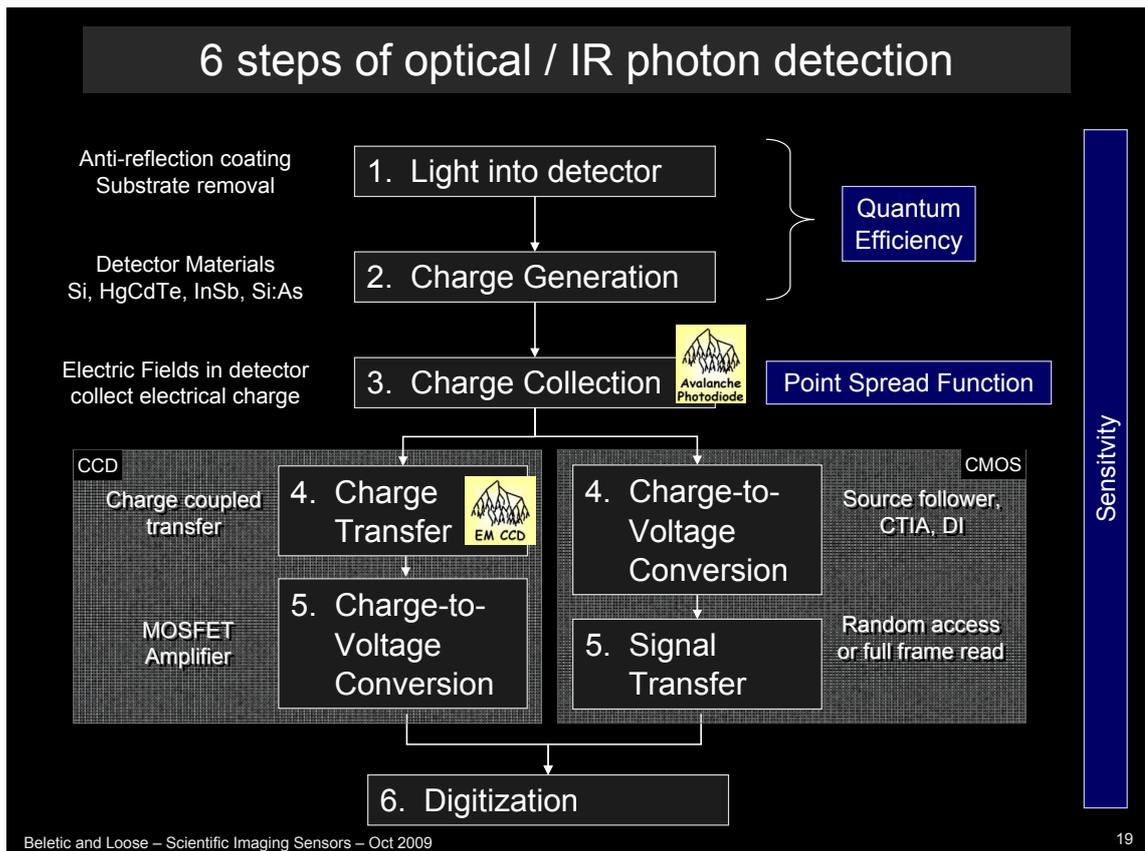


Figure 4.1: Functional diagram of an infrared light detection system [48].

large wavelength range. Some examples are listed in Figure 4.2. The thickness of the crystal will determine the probability of interaction of a photon with the crystal itself: this is called *quantum efficiency*. The thicker the crystal, the higher the probability of interaction. However, the increase in the thickness of the crystal is also accompanied by an increase in the sensitivity to ionizing radiation which can deposit enough energy along its trajectory through the crystal to make several electrons pass into the conduction band. The choice of the thickness of the crystal results therefore from a compromise between these two effects. After the conversion of the incident photon flux to charge carriers, it is necessary to collect them. This process is called the *collection of photo-generated charges*. In the absence of an electric field, the electrons and the holes created in the crystal will recombine. On the other hand, if an electric field is created at the crystal terminals, thus forming a plane capacitor, these carriers will move toward the electrodes and thus create an electrical signal. The principle is to apply a potential

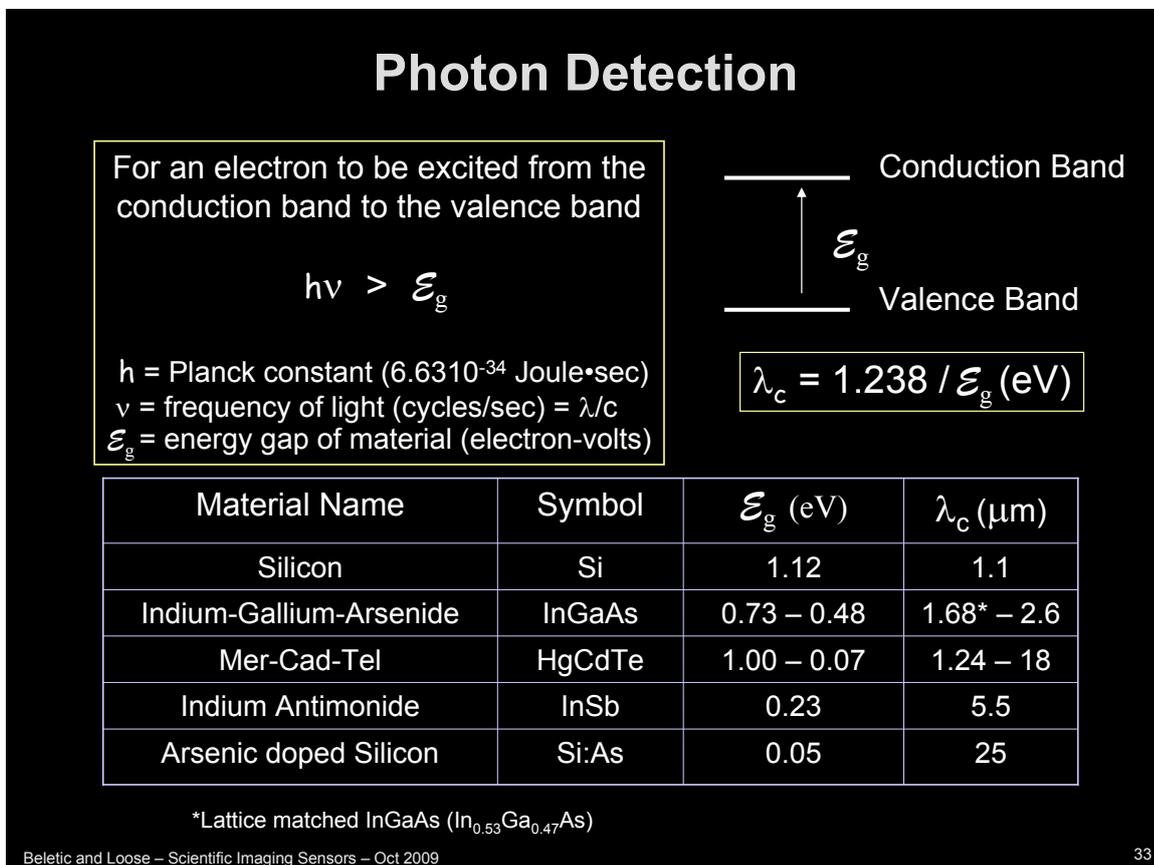


Figure 4.2: A table listing the energy to be supplied to electrons in order to pass the band gap for various semiconductor crystals with the associated cut-off wavelength [48]. As it is remarked for InGaAs crystal, the width of the band gap depends on the proportions between the different components; it holds for HgCdTe crystal as well. The sign “:” separates the base material from the “dopant”, so Si:As is a crystal of silicon doped by the addition of arsenic atoms.

difference  $\Delta V$  to the terminals of this capacitor thereby creating a uniform electric field in the volume. When creating charge carriers ( $q$ ), this electric field separates the carriers and sets them in motion towards the electrodes (according to their type), thus creating a variation of the potential in the capacitor. When all the carriers have been collected, the potential variation at the terminals of the capacitor is  $\Delta V = \frac{q}{C}$ , where  $q$  is the totality of the charges collected and  $C$  is the capacitance of the capacitor. In general, for a semiconductor, the resistivity of the crystal is not very high and the application of an electric field generates a current several orders of magnitude greater than the one which is desired to detect. In order to decrease this current, a *PN junction* is then used. A PN junction results from the contact of two crystals of the same type but having excesses of electrons (a material defined as *negatively doped* - N) or empty sites in the valence band, called *holes* (such a material is said to be *positively doped* - P). When a P- and a N-doped semiconductors come into contact, a *depletion region* or *Space Charge Region* (SCR) is produced, see Figure 4.3. By spontaneous diffusion of electrons and holes on either side of the PN junction, the holes migrate from P side to N side and vice-versa for the electrons. An electric field appears at the terminals of the junction. This field is opposed to any additional movement of the carriers: it acts as a potential barrier which increases the resistivity, thus creating a photodiode.

Figure 4.5 shows the evolution of the charge number (top), the electric field (center) and the voltage at the terminals of the PN junction (bottom). For a junction with no polarization, the contact between the two semiconductors causes the diffusion of charges on both sides of the

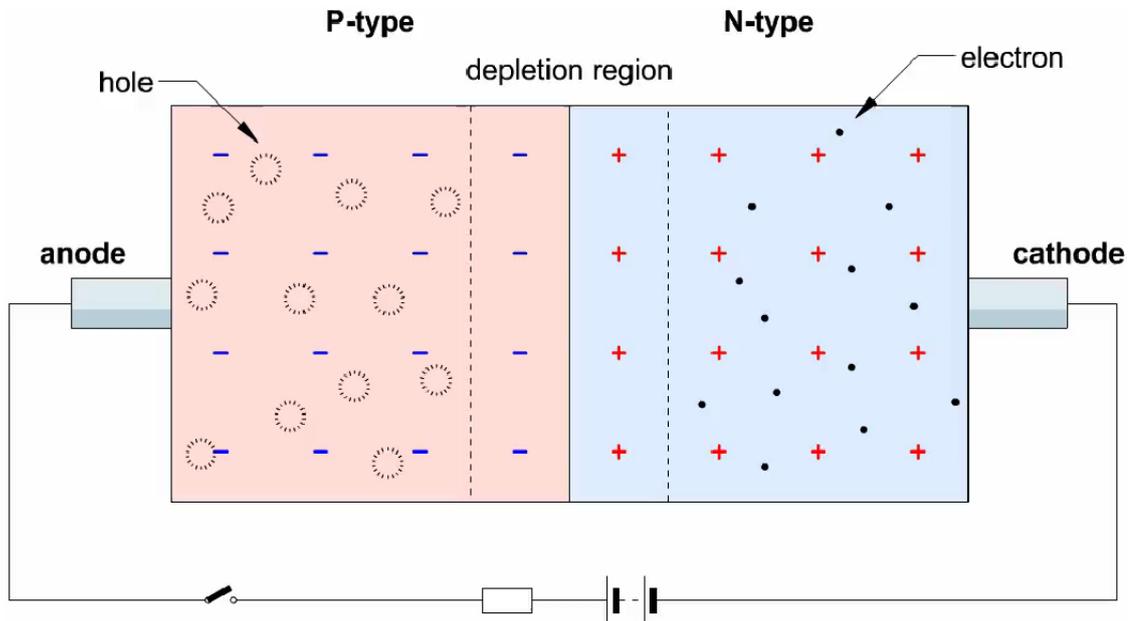


Figure 4.3: Picture showing the formation of the SCR according to the polarization at the terminals of the PN junction, in the case of two P and N doped crystals with the same concentration of charge carriers.

junction and the creation of a SCR. When the P side of the junction is brought to a negative potential with respect to the N side, then an inverse polarization (with a potential  $D_{sub}$  for the N side and  $V_{reset}$  for the P side, see Figure 4.4) is realized. In this case, an enlargement of the SCR is observed. If the voltage is sufficient, an electric field is thus created throughout the thickness of the semiconductor: the junction is said to be completely depleted and the charges created in the detector thickness can be collected. In order to convert a flux of photons into a number of charge carriers, a known voltage is first applied across the junction ( $D_{sub} - V_{reset}$ ) and the variation of the potential at one of the junction sides is measured as a function of time. In Euclid's case, we refer to the evolution of the potential starting from  $V_{reset}$ . This means that we measure the number of holes integrated since the beginning of the acquisition. If at a time  $t$  the potential measured at the P side of the junction is  $V_{reset}$ , at time  $t + 1$  it will be:

$$V_{t+1} = V_{reset} + N_{carriers} \cdot \frac{1}{C} \quad (4.2)$$

with  $N_{carriers}$  the number of electron-hole pairs generated by the arrival of photons in the HgCdTe crystal between  $t$  and  $t + 1$  and  $C$  the capacity of the junction. The quantity  $\frac{1}{C}$  is also called the *transimpedance gain* of the junction, which will be indicated with  $G_{SCA}$ . It should be noted that when the voltage  $V_t$  becomes equal to  $D_{sub}$ , the electric field in the detector vanishes and no more carriers can be collected (*saturation*). In order to return to the initial depletion level of the PN junction, it is sufficient to empty the photo-generated charges by re-applying the initial  $D_{sub} - V_{reset}$  voltage: this operation is called *reset* and restores the initial state of the reverse biased junction (Figure 4.5, in the middle). This is how a *charge-integrating detector* works. When the integrated incident photon signal has been converted into an electrical signal (the voltage variation at the terminals of the PN junction), it is necessary, after suitable amplification, to convert it into a digital signal. This analog-to-digital signal conversion is performed by an Analog-to-Digital Converter. The ADC has an input dynamic

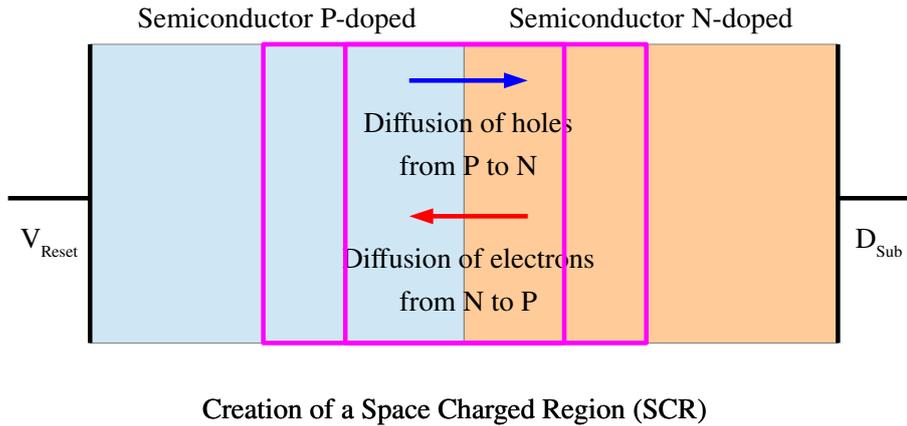


Figure 4.4: The SCR prevents the passage of the current: its width can be changed by modifying the potential difference applied to the terminals of the dipole.

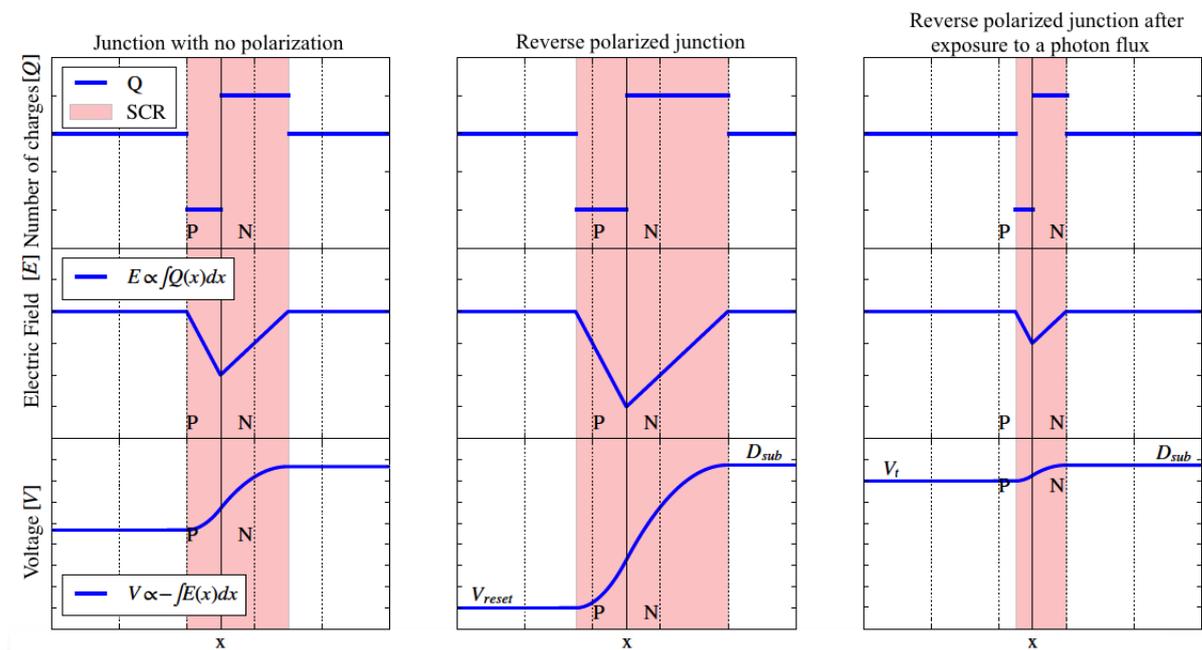


Figure 4.5: Representation of the number of charges, electric field and voltage within the PN junction (from top to bottom) for several operating modes: unpolarized, reverse biased, reverse biased with signal.

range (the voltage supplied from 0 to  $V_{max}$ ) and an output dynamic range (the number of bits on which the digitized value will be encoded). The output signal is usually expressed in ADU (Analogic to Digital Unit):

$$S_{ADU} = S_{volts} \cdot R_{ADC} \quad (4.3)$$

where  $S_{volts}$  is the measured voltage,  $S_{ADU}$  is the value of the digitized voltage. This conversion involves a quantification of the input signal and therefore introduces an error depending on its own theoretical resolution  $R_{ADC}$ :

$$R_{ADC} = \frac{V_{range}}{2^M - 1} \quad (4.4)$$

where  $V_{range}$  is the input dynamic range of the ADC and  $M$  the resolution in bits. In practice, the choice of the resolution of the ADC is generally imposed by the noise of the input signal. For example, a 0-5 V ADC that encodes on 16 bits will convert an input voltage of 0 V to 0 ADU and a voltage of 5 V to 65535 ADU ( $2^{16} - 1$ ): the resolution of such an ADC is  $R_{ADC} = \frac{5}{2^{16}-1} \approx 76.3 \mu\text{V}/\text{ADU}$ . It is unuseful to take an ADC with such a resolution if the signal noise at digitization is well above  $76.3 \mu\text{V}$ . There are also several limitations to the use of an ADC in an electronic circuit:

- *ADC limits*: when the input voltage is too low or too high, ADC limits are reached (respectively 0 or  $2^M - 1$ );
- *non-linearity*: the response of the SCE ADC to the range of measurable voltages is not always linear, especially at the end of the range; it is therefore preferable not to use it under these conditions.

We have thus described the entire detection chain, starting from the photo-generation of carriers using a semiconductor exposed to a flux of incident photons. The continuous measurement of the voltage is then amplified and digitized, and then it is possible to process this signal with a computer.

## 4.2 The pixel matrix of the IR detectors

We have just seen the operational principles of a “unitary” IR detector, i.e. a photodiode. If we now want to obtain an image, we will have to associate thousands of these photodiodes. In order to obtain the spatial distribution of the flux of incident photons, the detector is electrically subdivided into a 2D matrix of pixels (see Figure 4.6). This is commonly done in the visible range with sensors of the CCD or CMOS type. CMOS detectors are gradually replacing CCDs in digital photography applications. Like CCDs, CMOS sensors are in the form of a matrix of small-sized pixels (each with a few  $\mu\text{m}$  side) that will allow to sample a field of view. A major difference from CCDs is that each pixel contains a photodiode (used for photon-charge conversion and charge storage) and an amplifier that will convert the charges into a voltage in the pixel itself. Each pixel can thus be read independently without transferring charges between pixels as in CCDs. The reading is then non destructive and partial image readout is

possible (reducing the reading time). With this non destructive reading it is possible to make several successive measurements and to average them in order to reduce the noise. A matrix of visible CMOS pixels is indeed a matrix of Si photodiodes associated with amplification systems and with a row and column “switch” which allows to select the signal of the pixel to be directed to the output in order to be digitized (see Figure 4.7).

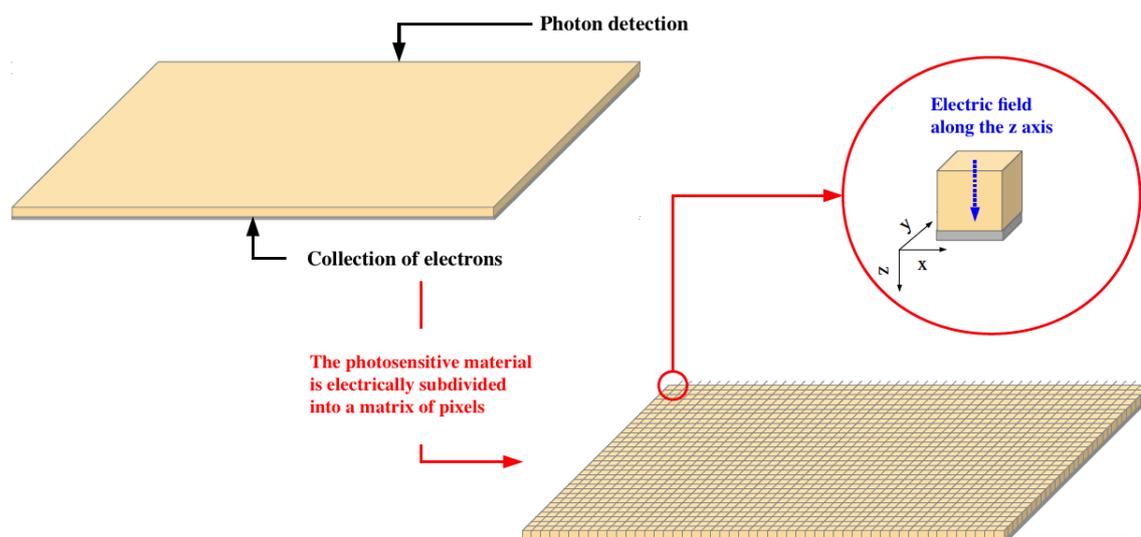


Figure 4.6: Representation of the electrical subdivision of a monolithic semiconductor by applying an electric field to each of the pixels.

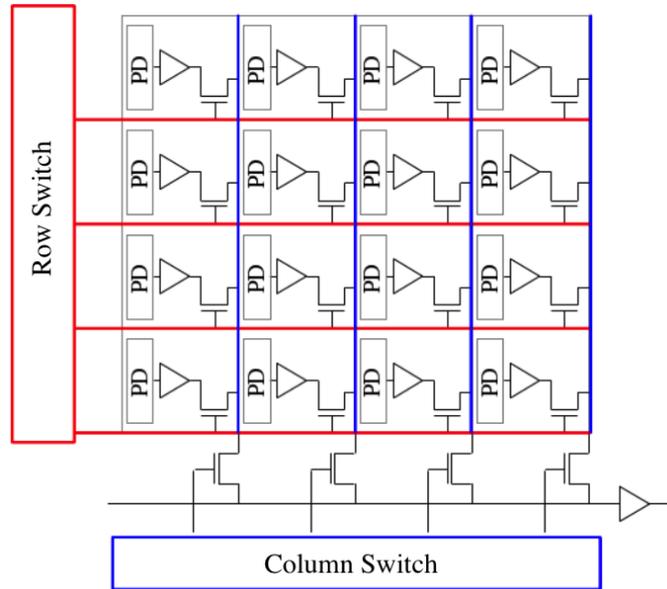


Figure 4.7: Row/column switch diagram of a CMOS type detector.

In order to read the variations of voltage across the photodiode and to reduce the read noise, the use of a signal amplifier in the pixel is necessary. A detector with a *follower transistor* (*source follower*, SF) in each pixel is called an APS (*Active Pixel Sensor*, mainly developed by Fossum since 1993 [49]). An APS detector pixel is shown in Figure 4.8. The photodiode is polarized by the use of the  $D_{sub}$  and  $V_{reset}$  voltages, while the presence of an SF transistor allows to transmit enough power through the readout bus; a *selection transistor* intervenes when the row/column switch selects the pixel for the reading process. The signal is then transferred to the readout bus for digitization. The APS reading technique is very similar to that of the visible CMOS pixels which is used for IR detectors of type H2RG. However, since silicon is only sensitive to photons in the visible (due to its band gap too large for NIR photons), it is necessary to use another type of low band gap crystal for the detection of infrared photons. Additionally, source follower and switches can only be implemented in silicon. A different material as detector implies a complex process of hybridization between detector matrix and front-end electronics.

### 4.3 Intrinsic properties of the detectors

We will now describe some general properties of the IR detectors highlighting the ones driving the measurement performances.

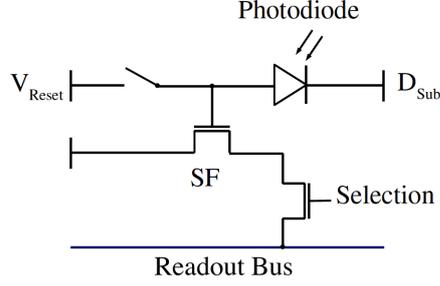


Figure 4.8: *Illustration of an APS detector pixel architecture.*

#### 4.3.1 Total noise

The total noise comes from two main components, the readout noise and the photon noise (Poisson). The readout noise is associated with each of the readings of the detector and is defined by a Gaussian distribution centered on 0 and with  $\sigma_{readout}$  standard deviation (expressed in electrons). The noise due to photons is described by a Poisson distribution with an average value of  $N$ , the number of photons expected during acquisition, and a fluctuation of  $\sqrt{N}$  around this average value. Between two successive measurements, these two components not being correlated with each other, the total noise results as the quadratic sum of the different contributions (see Equation 4.11). For each noise measurement, it is necessary to take into account these two components, which it is not possible to dissociate in flight and which depend on the used readout mode.

#### 4.3.2 Gain

The detection of the signal is achieved through three conversion steps and each of them is defined by a specific gain:

- the *quantum efficiency*, i.e. the conversion rate ( $\eta$  in carriers/photons) between incident photons and charge carriers generated in the pixels of the detector;
- the *transimpedance gain* of the sensitive part ( $G_{SCA}$  in mV/carriers), which is the conversion between the number of charge carriers accumulated in the pixel and the voltage variation across the pixel;
- the *PreAmp + ADC gain* of the SCE readout electronics, which is the conversion between the measured voltage and the digital signal to be recorded ( $G_{SCE}$  in ADU/mV).

The total gain in photons/ADU can be written as the inverse of the product of the conversion steps factors:

$$G = \frac{1}{\eta \times G_{SCA} \times G_{SCE}}. \quad (4.5)$$

The conversion gain in carriers/ADU is also defined as the inverse of the product of the transimpedance gain and the PreAmp gain:

$$G_{conv} = \frac{1}{G_{SCA} \times G_{SCE}}. \quad (4.6)$$

### 4.3.3 Non-linearity

Due to different physical and electronic effects, the detector response is not completely linear. In addition to the obvious non-linearities introduced by the ADC saturation, the non-linear response of the pixels is due to the increase in the pixels capacitance  $C_{node}$  by the accumulation of charge carriers which reduces the voltage across the pixels. This causes the correlation between the number of accumulated charges and the measured voltage to become non-linear, due to a decrease in the transimpedance gain ( $G_{SCA} = 1/C_{node}$ ). The non-linearity effects are shown in Figure 4.9. In green the response of a pixel reaching the limit of the ADC ( $2^{16} - 1 = 65535$ ) is represented, and in blue the same pixel is addressed, this time reaching the *Full Well Capacity* (FWC, see §4.3.5) and therefore being no longer able to collect electron-hole pairs. The ADC saturation is visible on the green curve by the slope change near the coding limit. The ramp in blue represents the normal response of a pixel with a hardly correctable non-linearity, which ends up with a strong non-linearity at the end of the ramp due to saturation of the pixel. In order to understand the response of a pixel over the entire dynamic range, it is therefore necessary to correct this non-linearity effect, especially at the end of the ramp.

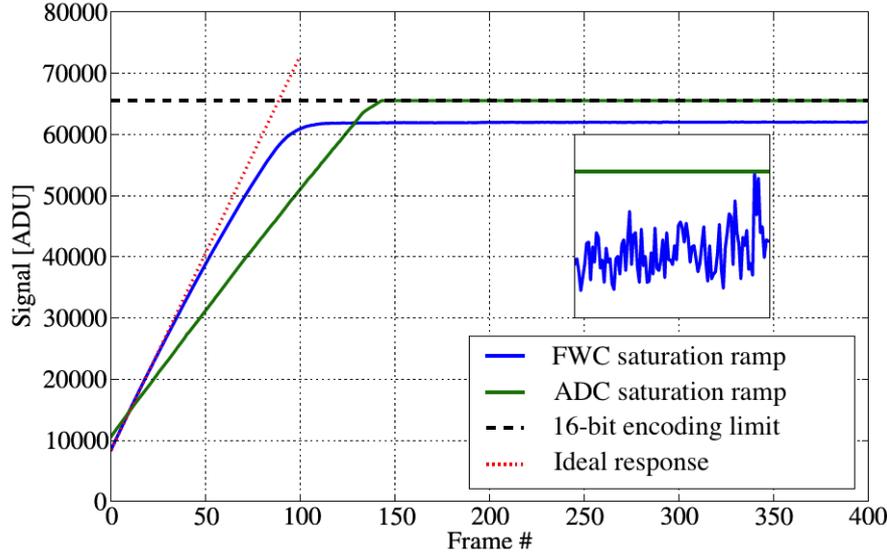


Figure 4.9: *Ideal response (red) and non-linear (blue) with saturation of one pixel of H2RG detector. In green a ramp of this same pixel reaching the saturation of the ADC (made with another acquisition with a gain of 21 dB instead of 18 dB to observe ADC saturation) is represented.*

#### 4.3.4 Inter-Pixel Capacitance (IPC)

Each pixel is electrically separated from its neighbors by the application of an electric field along the PN junction, but there is a correlation between the signal measured on a pixel and its nearest neighbors. This effect can be interpreted as a capacitance between the pixels [50]. The *inter-pixel capacitance* is illustrated in Figure 4.10 with  $C_{ip}$  the inter-pixel capacitance and  $C_{node}$  the capacitance of the pixel node. As a result of these parasitic capacitance, the original signal located in one pixel is lower than expected and the remaining signal appears on the neighbouring pixels.

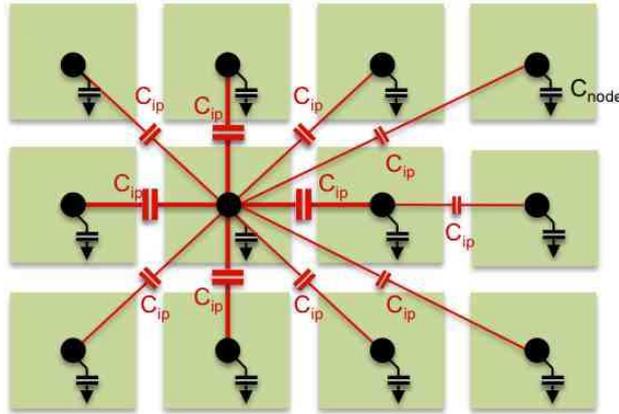


Figure 4.10: Illustration of the inter-pixel capacitance effect [51].

#### 4.3.5 Full Well Capacity (FWC)

When the SCR of the pixels of the detector is minimal, it is said that the *Full Well Capacity* or saturation of the pixels is reached. This limit is defined by the number of carriers which it is possible to collect for a given detector polarization.

#### 4.3.6 Persistence

The HgCdTe crystal of the detector has intrinsic defects arising from the crystal growing process. These defects are responsible for the *persistence* effect, which consists of a trapping of the charges inside the pixel; these charges will then be released over time during subsequent acquisitions [52] [53]. This phenomenon of trapping/de-trapping charges in the detector is responsible for the presence of a remanent image after the exposure of the detector to a high photon flux.

#### 4.3.7 kTC noise

*kTC noise* is a thermal noise associated with capacitors, which do not dissipate power or produce noise: it is the process of resetting a capacitor that introduces noise, and that is kTC

noise. Each time a capacitor is reset, a random variation is introduced on the capacitor voltage, since at temperatures above absolute zero, free electrons in conductors exhibit random motion. If we examine the voltages on a capacitor in a succession of resets, we will find slightly different values for the voltages and for the stored charge. The standard deviation of those values is the *kTC noise*. A simple theoretical model for this noise can be studied considering a capacitor being reset by shorting it with a resistor in an RC circuit. The noise bandwidth of the RC circuit is  $1/(4RC)$ , so that if we substitute that bandwidth into the *Johnson noise* formulas<sup>1</sup>, the resistance cancels out, yielding a simple expression for the variance of the voltage:

$$\sigma_V^2 = \frac{kT}{C}. \quad (4.7)$$

We can also express this noise in terms of the variance of the electrical charges ( $\sigma_q^2$ ), or the equivalent variance of the number of carriers on the capacitor ( $\sigma_N^2$ ):

$$\sigma_q^2 = (\sigma_V C)^2 = kTC \quad (4.8)$$

$$\sigma_N^2 = \left(\frac{\sigma_q}{e}\right)^2 = \frac{kTC}{e^2} \Rightarrow \sigma_N = \frac{\sqrt{kTC}}{e} \quad (4.9)$$

where  $e$  is the electron charge. In detector readout kTC noise is responsible for the variation of the content of the first frame (*baseline*) after reset. kTC noise can be eliminated by CDS, Fowler and UTR sampling (see Figure 4.11).

## 4.4 The H2RG detectors of the NISP instrument

In this part, the technology used for the pixels of the H2RG detectors will be explained. The readout procedure of each pixel of the H2RG detectors will be discussed in detail, in particular by explaining the concept of acquisition channel and the different detector readout modes. For the Euclid mission, the wavelength range of interest is in the near infrared range ( $0.920 \mu\text{m} < \lambda < 2 \mu\text{m}$ ). As can be seen in Figure 4.2, some materials are composed of several elements (such as InGaAs and HgCdTe) and their range of detectable wavelengths is variable. These are called *adjustable band materials*. In our case, the choice is based on  $\text{Hg}_{1-x}\text{Cd}_x\text{Te}$  (Mercury-Cadmium Telluride) detectors which detect photons in this wavelength range. According to the proportion  $x$  between the two elements Hg and Cd in the crystal, the energy of the band gap (and therefore the detectable wavelength range) varies according to the following equation [55]:

$$\epsilon_g = -0.302 + 1.93x - 0.81x^2 + 0.832x^3 + 5.35 \cdot 10^{-4}T(1 - 2x) \quad (4.10)$$

with  $\epsilon_g$  in eV,  $x$  is the fraction of Cadmium in the crystal and  $T$  is the operating temperature of the detector. This relationship is illustrated in Figure 4.12, which shows the evolution of

---

<sup>1</sup>The *Johnson noise*, also known as *thermal noise* or *Nyquist noise*, is generated in all resistors by thermal agitation of the charge carriers at equilibrium; the value of the Johnson noise in terms of voltage variance is given by  $\sigma_V^2 = 4kTR\Delta f$ , where  $k$  is the Boltzmann constant ( $1.38 \times 10^{-23}$  J/K),  $T$  the temperature of the resistor (in K),  $R$  the resistance (in Ohms) and  $\Delta f$  the noise bandwidth (in Hz).

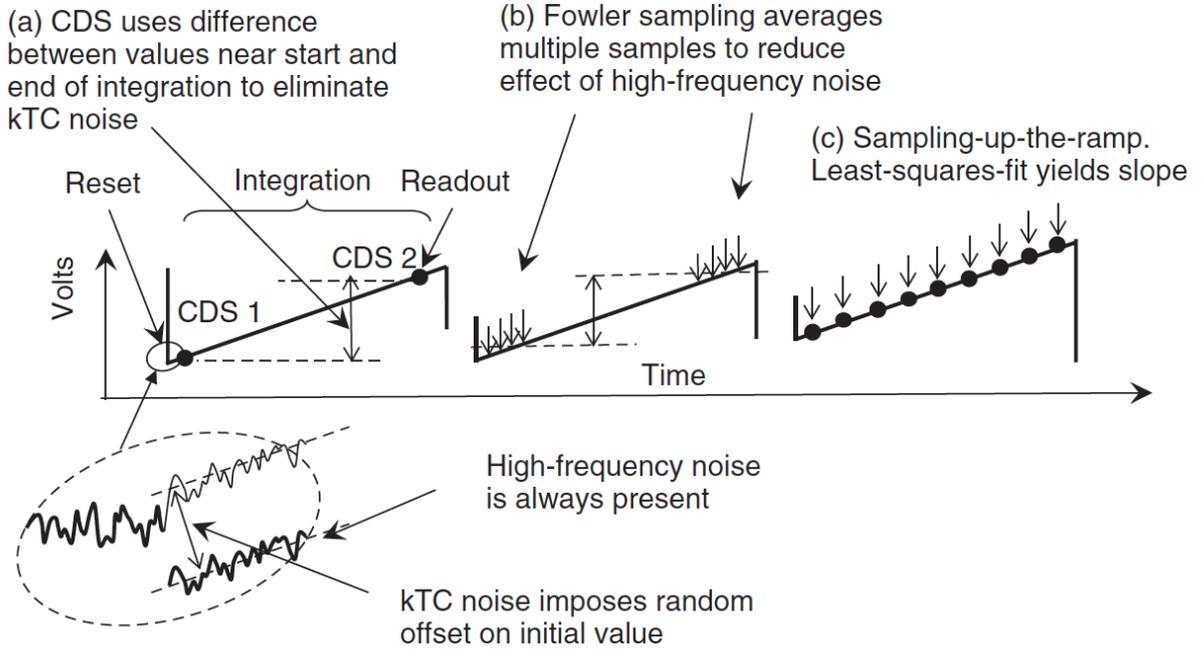


Figure 4.11: *Illustration of sampling methods to reduce or eliminate kTC noise. (a) Correlated double sampling (CDS). (b) Fowler sampling. (c) Sampling Up The Ramp [54].*

the band gap energy  $\epsilon_g$  and of the cut-off wavelength  $\lambda_c$  as a function of the fraction  $x$  of Cd in the crystal with a detector at 77 K (in blue) and at 300 K (dashed line). In the case of the detectors of the NISP instrument, we have  $\lambda_c = 2.3 \mu\text{m}$ , corresponding to  $x = 0.54$ . The amount of energy to be supplied to create an  $e^-$ /hole pair,  $\epsilon_g$ , is small and thermal agitation at ambient temperature is enough to create thousands of them, creating a parasitic current called “dark current”. With this cutoff wavelength, the detectors can only work at temperature around 100 K or below.

#### 4.4.1 The detection system: Silicon Chip Sensor (SCS)

The entire detection chain system (the SCA and the SCE, connected by a flexible cable at cold temperature) constitutes what is called an SCS (Silicon Chip Sensor) and is represented in Figure 4.14. A schematic diagram of the full chain is given in Figure 4.13.

#### 4.4.2 The Sensor Chip Assembly (SCA)

The CMOS technology, used for cameras in the visible, allows to integrate the silicon photodiode into the circuit with the amplification transistors as well as the switching chain, which is also made of silicon. In the case of infrared, in order to associate the HgCdTe detection material with the transistors allowing the readout of the voltages, a so-called *hybrid architecture* is adopted. This is the solution adopted by Teledyne Imaging Sensors (TIS) Hawaii  $2k \times 2k$

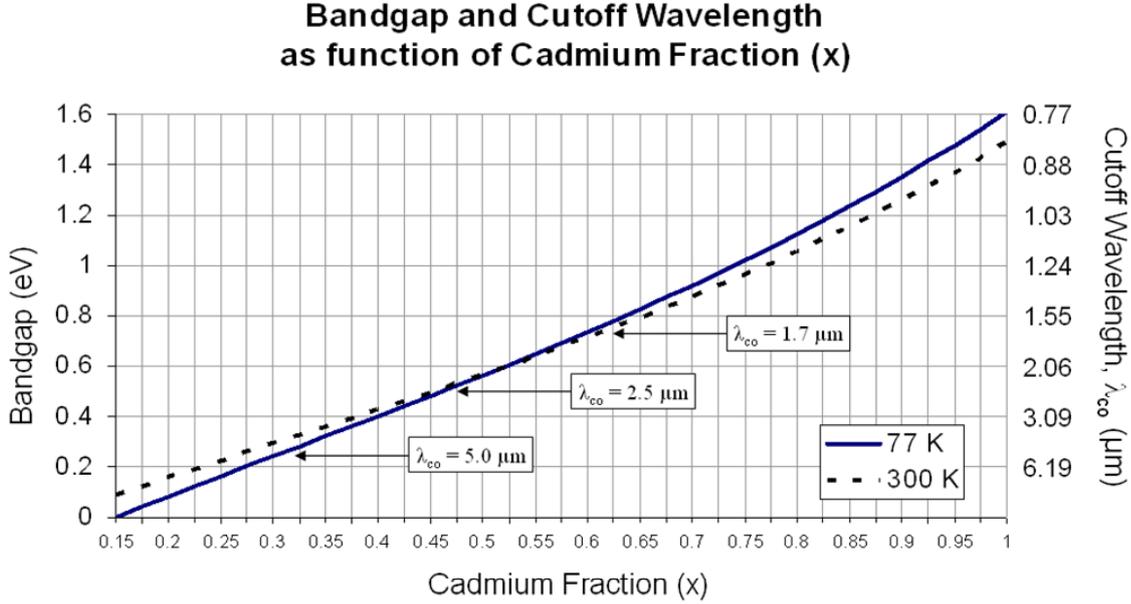


Figure 4.12: Variation of the band gap energy and of the cut-off wavelength of a  $\text{Hg}_{1-x}\text{Cd}_x\text{Te}$  detector as a function of the fraction of Cadmium  $x$ .

with Reference pixels and Guide mode (H2RG) [57]. The architecture of these detectors is shown in Figure 4.15. These hybrid detectors mainly consist of a  $\text{HgCdTe}$  crystal for collecting charges, composed of more than 4 million pixels ( $2048 \times 2048$ ). This matrix is then coupled to a silicon chip called “multiplexer” by means of indium bumps. This multiplexer includes all the already described source follower and switches necessary to read out the 4 million pixels. The clocking for the sequential readout of the pixels is ensured by a 100 kHz clock. However, in order to lower the reading time, the pixel matrix is divided in 32 “channels” which are read in parallel.

#### 4.4.3 The readout electronics: Sensor Chip Electronics (SCE)

In order to control the readout circuit (multiplexing) of the SCA, an Application Specific Integrated Circuit (ASIC) is used. It manages all aspects of SCA operations as well as the digitization of the signal coming from the detector. The SIDECAR ASIC [59], to be known as SCE, may be configured to read the detector signal using several identical channels with parallel outputs. In the so-called *Euclid configuration*, we use 32 channels: this corresponds to a subdivision of the acquisition into 32 sub-matrices of  $64 \times 2048$  pixels (see Figure 4.16), leading to a readout time of 1.41 s for the entire matrix. The main functions of the SCE for our use of the detectors are summarized in Table 4.1. This circuit will mainly generate all the clocks necessary for the operations of the SCA and ensure the digitization of the 32 video outputs of the SCA. Note also the presence of a microcontroller which allows the SCE to perform operations on the acquired images. So, every 1.41 s the SCE allows to read the voltage

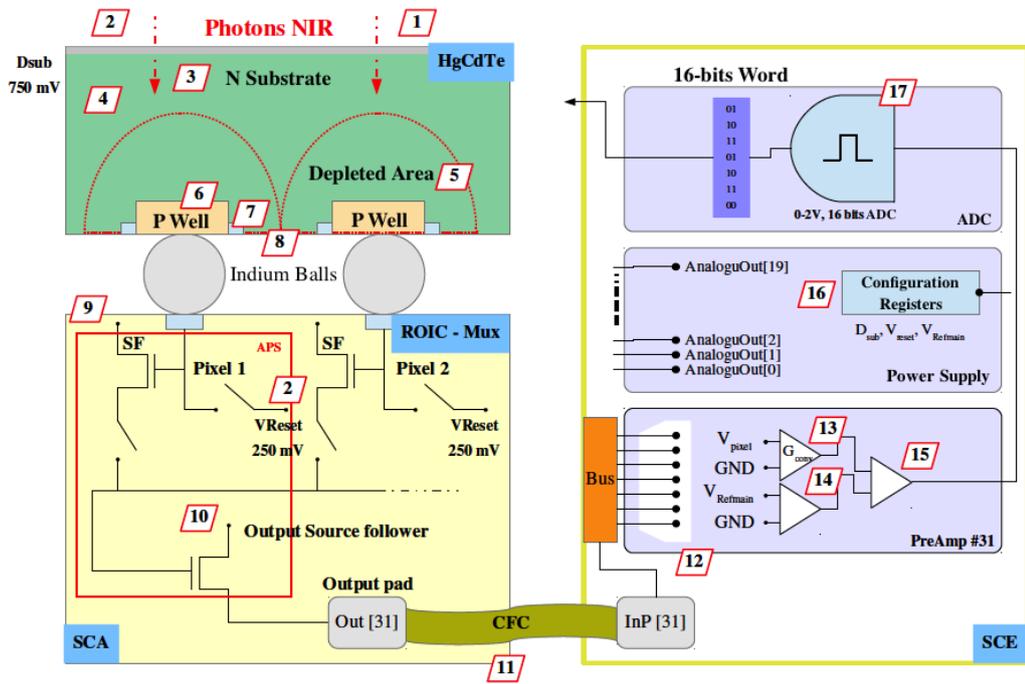


Figure 4.13: Representation of the whole architecture of the H2RG detectors [56].

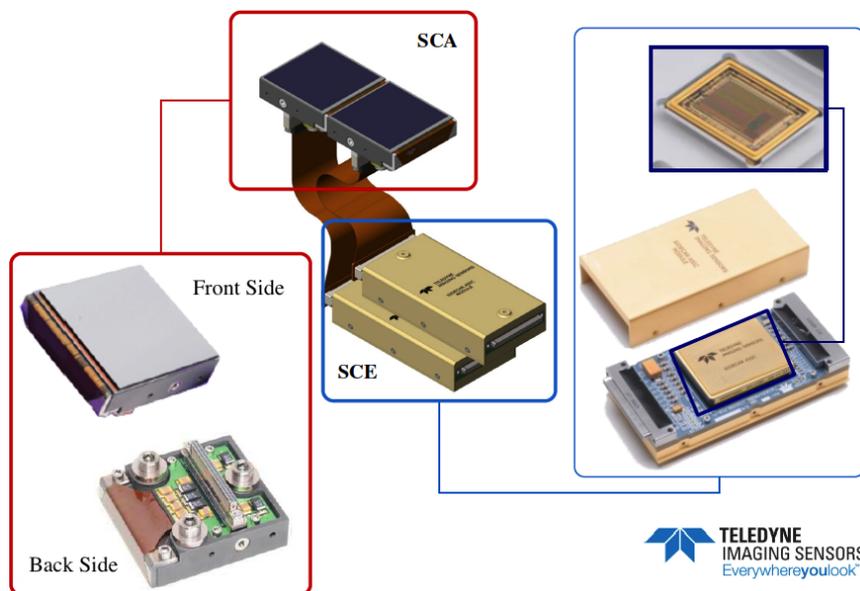


Figure 4.14: Image of the SCS chain with the Sensor Chip Assembly (SCA), the Sensor Chip Electronics (SCE) and the CFC (flexible linking circuit).

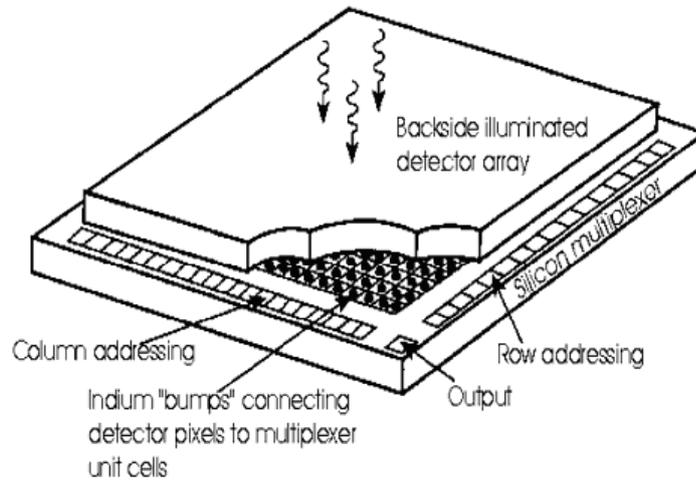


Figure 4.15: *Illustration of the hybrid architecture of a H2RG detector [58].*

present on the P side of the PN junction of the pixels. Each of the pixels of the detector being encoded on 16 bits, the total size of an image (without additional information) is 8 MB.

---

#### Teledyne SIDECAR ASIC Options

---

32 acquisition channels for the detector

1 additional channel for reference channels output/temperature measurements

Scanning frequency up to 500 kHz with resolution of 16 bits per channel

Adjustable gain of the preamplifier from 0 to 27 dB by steps of 3 dB

20 programmable polarization voltages/currents (including  $D_{sub}$  and  $V_{reset}$ )

16-bit microcontroller for acquisition operation

---

Table 4.1: Table of configuration options available for the ASIC developed by TIS.

## 4.5 Readout modes for H2RG detectors

Regardless of the selected operation mode, the SCE will read and scan all images at maximum readout rate. This is necessary for the thermal stability of the system (so that the detector is always in operation, therefore absorbing a stable amount of power). Similarly, during periods when observation is interrupted (change of filters, stabilization periods, change of field of view), the system is continuously active. For this reason, the SCE allows the acquisition of an image which is transferred to a processing electronics (called *Warm Electronics* in the Euclid case) or the same acquisition without transfer. The successive readouts of the detector are non-destructive. During an acquisition, a measurement of the integrated signal since the beginning of the exposure is continuously made. Several strategies for continuous readout of

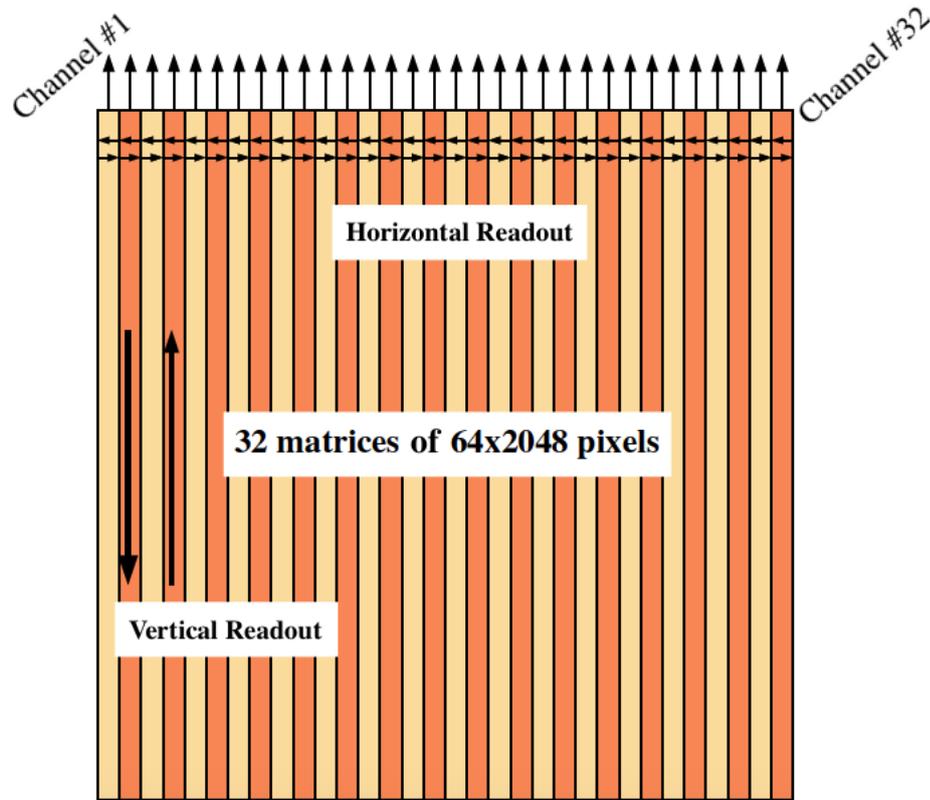


Figure 4.16: Readout setup scheme by means of an H2RG detector: the arrows at the top represent the outputs of each of the 32 channels, the other arrows indicate the possible directions of reading (reading from bottom to top and from left to right for each channel or vice-versa).

the detector can be defined by the operation modes of the SCE. They are described hereafter by taking the example of an isolated pixel, but the processing of the whole matrix is the simple iteration of these operations. The choice of one or the other of these strategies will depend on many factors, in particular the amount of data to be acquired, stored and processed.

#### 4.5.1 The correlated double sampling (CDS)

In order to obtain a value of the integrated signal during an exposure of duration  $t_{exp}$ , the so-called *double sampling* (CDS for *Correlated Double Sampling*) is done, that is to say the signal is measured immediately after the reset and just at the end of the exposure. This is shown in Figure 4.17, where each bar represents a reading of the detector signal (one every 1.41 s). The red bars represent the “reset” images, not transferred: in this case  $V_{reset}$  is constantly applied to the pixel, which is therefore in permanent reset (no signal accumulation). The blue bars represent a reading of the detector signal which is then transferred to the electronics (“read”). The green bars indicate a reading which is not transferred (“drop”). The dashed

black lines represent the  $t_{exp}$  exposure time. This readout mode comprises only two readings, one at the beginning and the other at the end of acquisition for an exposure. The data obtained are reduced to a single value per pixel  $\Delta S = S(t) - S(t_0)$ , where  $S(t)$  is the measurement of the signal at time  $t$  in ADU. The uncertainty on the signal  $\Delta S$ ,  $\sigma_{CDS}$ , is then related to the readout noise and to the Poisson noise for the statistics of the incident photons:

$$\sigma_{CDS} = \sqrt{N_{carriers} + 2 \times \sigma_{readout}^2}. \quad (4.11)$$

This readout noise, inherent to each of the readings of the detector, arises from multiple causes (thermal noise of the output transistor, quantization noise of the encoder or even interference on the transmission lines). The order of magnitude of this noise for a H2RG detector under the Euclid conditions is  $\sigma_{readout} \approx 10 e^-$ , after correction from the reference pixels (see §4.6.1). Considering a null signal ( $N_{carriers} = 0$ ) and a typical readout noise of  $10 e^-$ , the CDS noise is  $\sigma_{CDS} = 14.1 e^-$ . It should be noted that for illuminations such as those expected during the Euclid survey, which are of 100 up to 1000  $e^-$  over the duration of the exposure, with  $t_{exp}$  ranging from 100 s (photometry) to 500 s (spectroscopy), this measurement method can lead to errors of more than 10%.

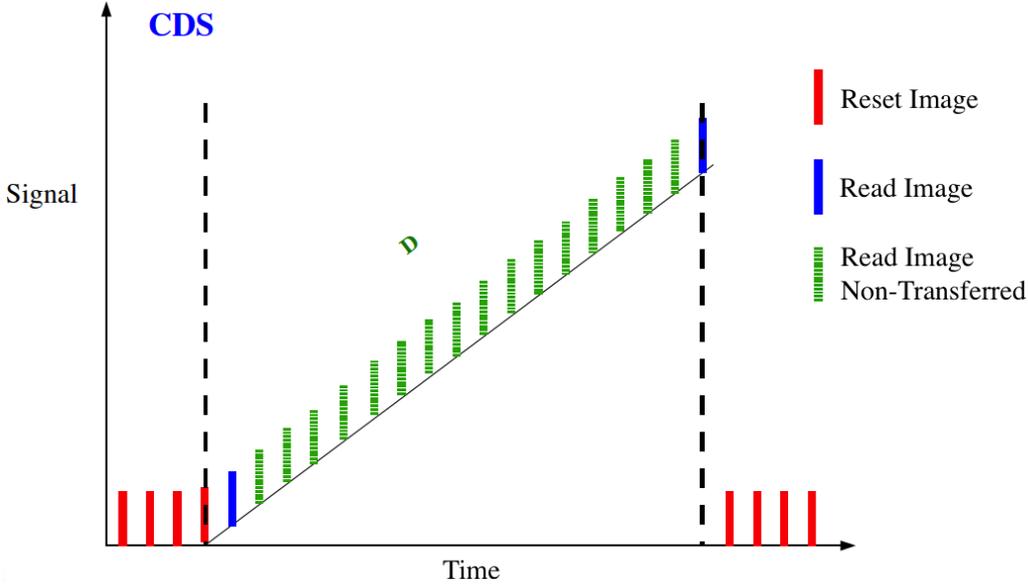


Figure 4.17: *Illustration of CDS readout mode.*

#### 4.5.2 Fowler-M acquisition

In order to reduce the readout noise associated with each reading of a pixel, the most commonly used technique is *Fowler-M* type acquisition. Instead of an acquisition in CDS mode, where only two images separated by a  $t_{exp}$  time interval are taken, in Fowler-M mode

the mean values of two groups of  $M$  images at the beginning and at the end of the ramp are used. If, for CDS, the integration time is equal to the exposure time (i.e. the time between the first and last frame), this is not the case for a Fowler-M type acquisition. Here we have:

- the *exposure time* which corresponds to the time separating the two groups of averaged images:  $t_{exp} = (N_{images} - N_{Fowler}) \cdot \delta t$  (grey lines in Figure 4.18);
- the *integration time* which corresponds to the time during which the detector accumulates photo-generated charge carriers:  $t_{int} = N_{images} \cdot \delta t$  (black lines in Figure 4.18).

The noise for a measure  $N$  of  $M$  averaged images is given by:

$$\sigma_N = \frac{\sigma_{readout}}{\sqrt{M}} \quad (4.12)$$

and therefore the accuracy of the Fowler-M measurement is:

$$\sigma_{Fowler-M} = \sqrt{\frac{2 \times \sigma_{readout}^2}{M}} = \frac{\sigma_{CDS}}{\sqrt{M}}. \quad (4.13)$$

The evolution of the theoretical Fowler-M noise as a function of the number of averaged images is illustrated in Figure 4.19. Note that the Fowler-1 corresponds to the CDS mode. The Fowler-M mode is useful for reducing the contribution of the readout noise to the measurement of the signal of the pixels.

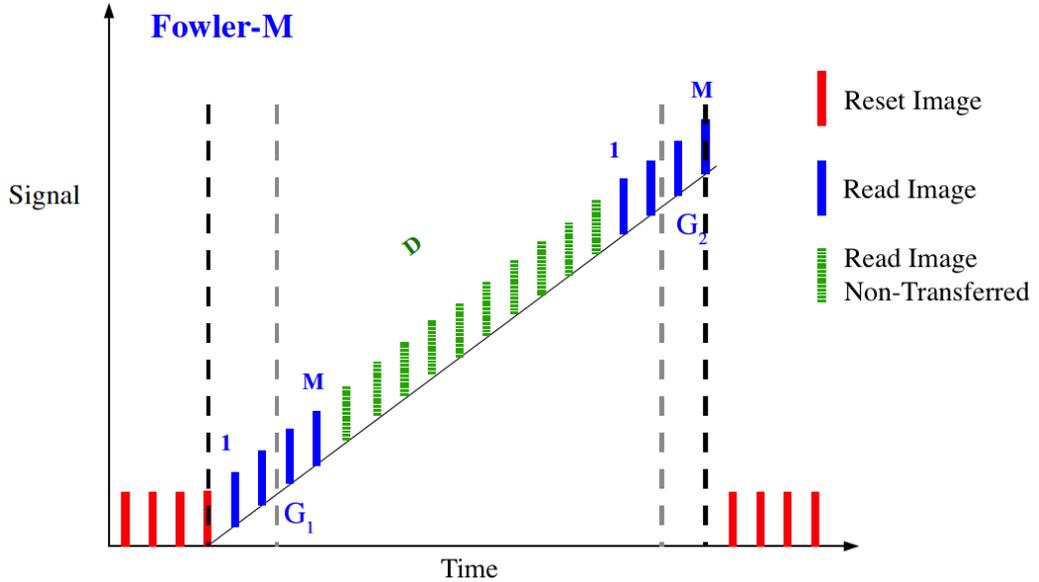


Figure 4.18: *Illustration of Fowler-M readout mode.*

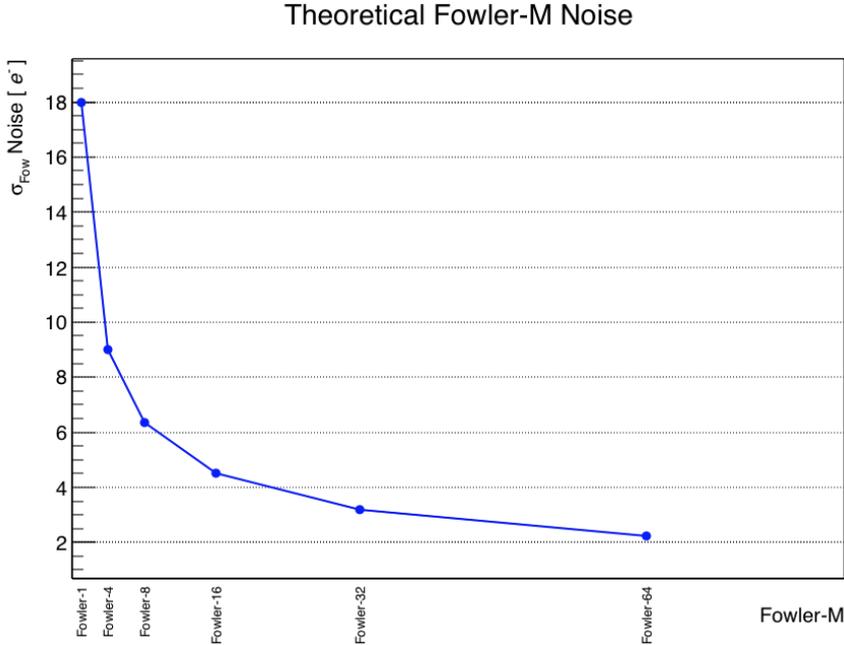


Figure 4.19: Noise gain theoretically estimated using Fowler- $M$  mode to average  $M$  images at start and end of the integrated signal ramp.

### 4.5.3 The acquisition of the “Up The Ramp” type (UTR)

In this type of acquisition, the detector data will be read and transferred at every reading (every 1.41 s, see Figure 4.20). The error between two successive measurements is the quadratic sum of the readout error and the Poisson error due to the number of incident photons, so that two successive measurements are correlated because of the Poisson error. The flux may be estimated, for example, by a linear fit of the points of the ramp. The amount of data to be stored and processed is considerable, but this method allows to detect the possible singularities that may occur during a ramp, in particular the impact of a cosmic ray.

### 4.5.4 The acquisition of Multiaccumulated type (MACC)

In order to combine the advantages of the UTR and Fowler- $M$  modes, a *Multiaccumulated* (MACC) mode can be used. This acquisition mode consists in reading images in groups of  $M$  read images, separated by  $D$  non-transferred images (see Figure 4.21). Thus, an acquisition in MACC mode can be defined by the number of images of a group  $M$ , the number of non-transferred images  $D$  and the number  $N$  of groups. We will simplify the notation of this readout mode by MACC  $[N, M, D]$ . It is important to note that for any readout mode, if the value of the signal is plotted as a function of time, a ramp is obtained whose slope is proportional to the photon flux. This flux information is the one to be obtained with the NISP instrument and its determination is based on a knowledge of the intrinsic effects of the detector physics.

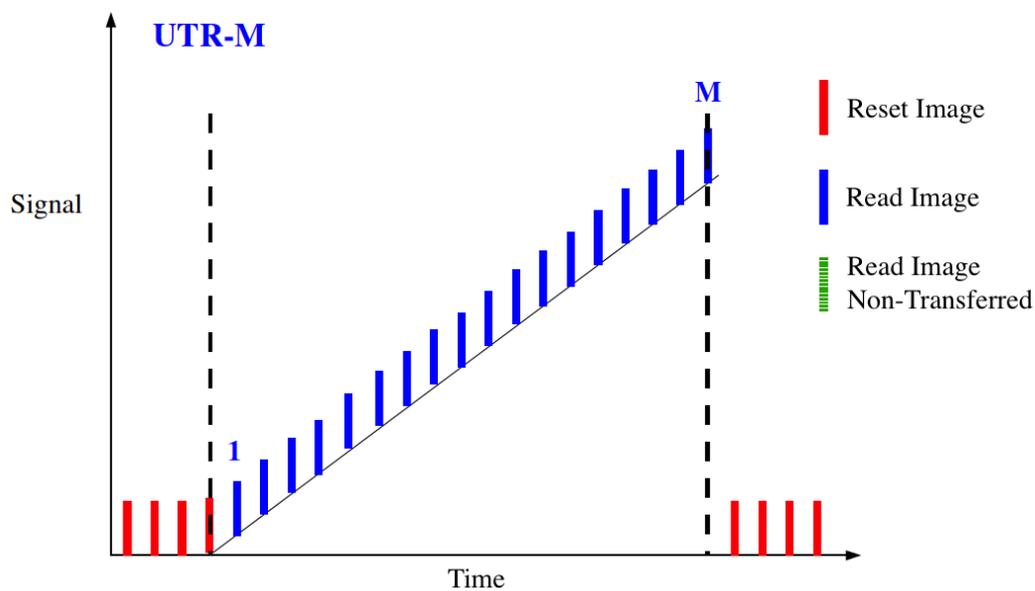


Figure 4.20: Illustration of the UTR-M readout mode.

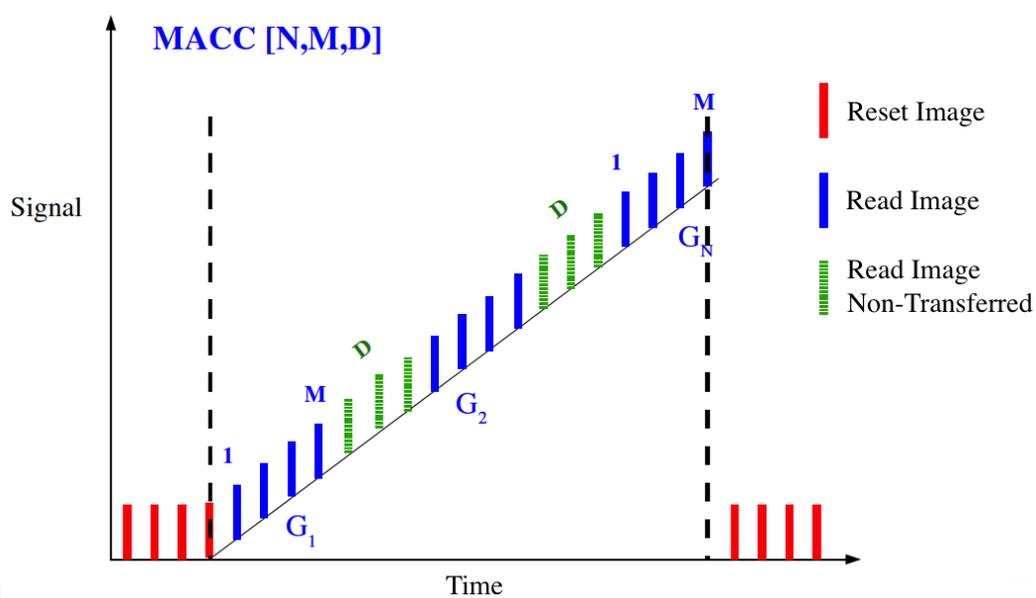


Figure 4.21: Illustration of the MACC  $[N,M,D]$  readout mode.

### 4.5.5 The choice of the readout modes for NISP

We have seen that the performance of detector noise is related to the readout mode used. Numerous constraints will influence the choice of the readout mode for the H2RG detectors, some based on the performance specifications mentioned in the previous sections, others imposed by spatial constraints and data processing on ground:

- when using the detector in CDS mode (the simplest) the noise can reach  $16 e^-$ ; in order to reduce this noise to less than  $9 e^-$  and  $13 e^-$  in photometric and spectroscopic mode respectively, groups of averaged successive images must be used either through Fowler-M mode or MACC  $[N, M, D]$  mode;
- the NISP instrument has a telemetry constraint of 250 Gbits/day, which requires that only few pieces of information per pixel (depending on the information coding) can be transferred to the ground; for most readout modes, sending all images exceeds the transfer capacity allocated to the instrument: it is therefore necessary to implement an in-flight data processing algorithm which directly calculates the flux received by the pixels;
- the detectors of the instrument are subjected to transient events such as cosmic rays, that will have to be detected and signaled; this adds further constraints to the functions of in-flight data processing.

### 4.5.6 The NISP MACC readout modes

For the NISP instrument, an optimization of the MACC readout modes is performed with respect to the exposure times imposed for each of the MACC  $[N, 16, D]$  modes. Equation 4.14 optimizes the MACC mode with respect to the exposure time for each combination of parameters  $N$ ,  $M$  and  $D$ :

$$t_{exp} = [N \times (M + D) - D] \times t_{image} \quad (4.14)$$

with  $t_{image}$  the time between two successive images (1.41 s). The time required for the in-flight electronics to process the data is about 7 seconds for each group ( $N$ ) acquired. Therefore, a number of non-transferred images ( $D$ ) of 5 allows time for the in-flight electronics to process the data. Considering these implementation requirements and the exposure times defined for each of the modes, the MACC readout modes described in Table 4.2 were chosen. The fact of

Observation mode	Groups (N)	Images (M)	Drops (D)	Time budget (s)
Spectroscopy	15	16	11	560
Y-band photometry	4	16	7	105
J-band photometry	3	16	6	83
H-band photometry	3	16	5	82

Table 4.2: Readout modes adopted for each of the observation modes of the NISP instrument.

having several groups in the ramp will allow to identify singular events in flight. Figure 4.22 gives an example of a simulated flux of  $2 e^-/s$  in MACC spectroscopic mode [15,16,11]. At the top, the UTR mode signal is shown. At the bottom, the spectroscopic mode illustrates the gain in noise obtained by the averaging of the 16 images of the groups, the error bars representing the associated readout errors. We can see the advantage of using the average of group images on the determination of the flux.

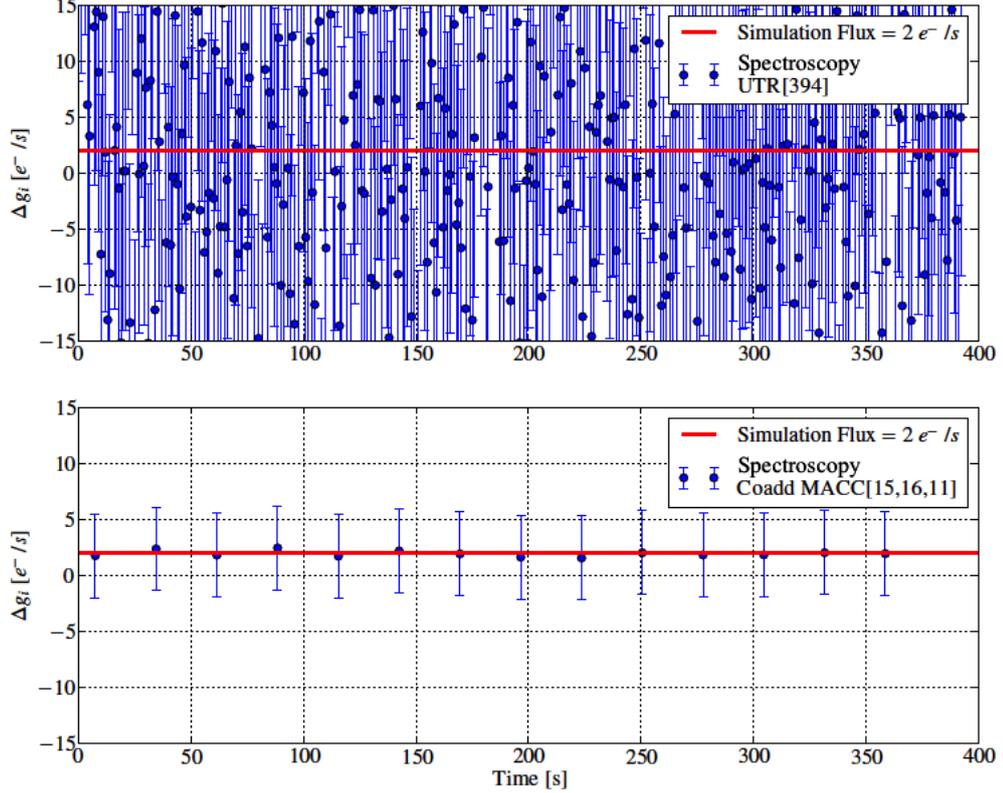


Figure 4.22: *Top: Difference of signal between successive images not grouped in UTR spectroscopic mode [394]. Bottom: Difference of successive groups mean in MACC spectroscopic mode [15,16,11].*

## 4.6 Readout noise reduction

When the detector readout mode performed is CDS, the readout noise is well above the specifications. We have seen that it is possible to average successive images in order to reduce the readout noise. In practice, the decrease in readout noise as a function of the number of averaged images decreases less rapidly than expected as a function of  $M$  as shown in Figure 4.23. We can deduce a limit, illustrated by the asymptote followed by the red curve, which shows that the maximum reduction of the readout noise is obtained by using an average of 16 successive

images, going from an average noise in Fowler-1 mode at  $16.6 e^-$  to  $6.7 e^-$  in Fowler-16 mode. It is thus possible to average 16 images per group to achieve optimum noise reduction. The method reduces the number of points available in order to fit the ramp but also visibly reduces the errors made on each of the groups. This is also the case with a MACC  $[N, 16, D]$  mode that has the same characteristics as a Fowler mode in terms of noise, but can additionally detect singularities in the ramp.

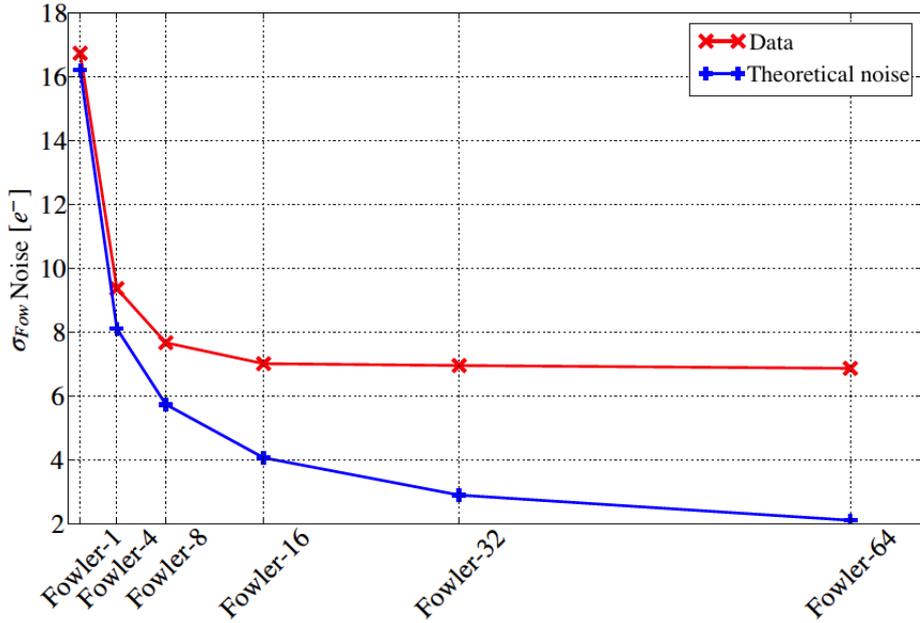


Figure 4.23: Gain in readout noise using Fowler- $M$  mode to average  $M$  images at start and end of ramp. In red, the Fowler noises obtained with a measurement of dark current (Poisson noise negligible). In blue, the theoretical Fowler noise for a given readout noise of about  $16 e^-$ .

#### 4.6.1 The common mode correction

In addition to the readout noise, previously defined, the subdivision of the SCA readout with the 32 channels of the SCE will result in the appearance of a new noise source. Indeed, each of the transistors of the channels will have its own variation of output level (step 10 in the SCS chain of Figure 4.13). This means that there is an offset between one channel and another due to this variation in the output level. This phenomenon is more visible when one is interested in the difference of successive images. This “common mode” noise can be observed in Figure 4.24, at the top left, with the appearance of 32 distinct vertical bands on the CDS noise (standard deviation of the differences between two successive images), so that, when we plot the distributions of the pixels of each channel, represented by the 32 red histograms, these distributions show mean values that are not centered on the same value. For the correction of the variations in the output voltages of the transistors of the SCE, and therefore of this

significant shift between each channel, there are reference pixels on the matrix, which are the four columns and rows defining the contour of the matrix. These pixels are at all points electronically identical to the others, but are not coupled to the sensitive part and therefore their measurement does not depend on the flux of incident photons. On the other hand, these pixels are electrically sensitive to channel variations, for example. Reading these pixels and subtracting their value to the image pixel on a channel by channel basis, greatly reduces the readout noise. This technique is known as *reference pixels correction*. It can be seen at bottom of Figure 4.24 that this correction effectively reduces the noise, with the histograms of the CDS noise of each channel that are well centered on the same value.

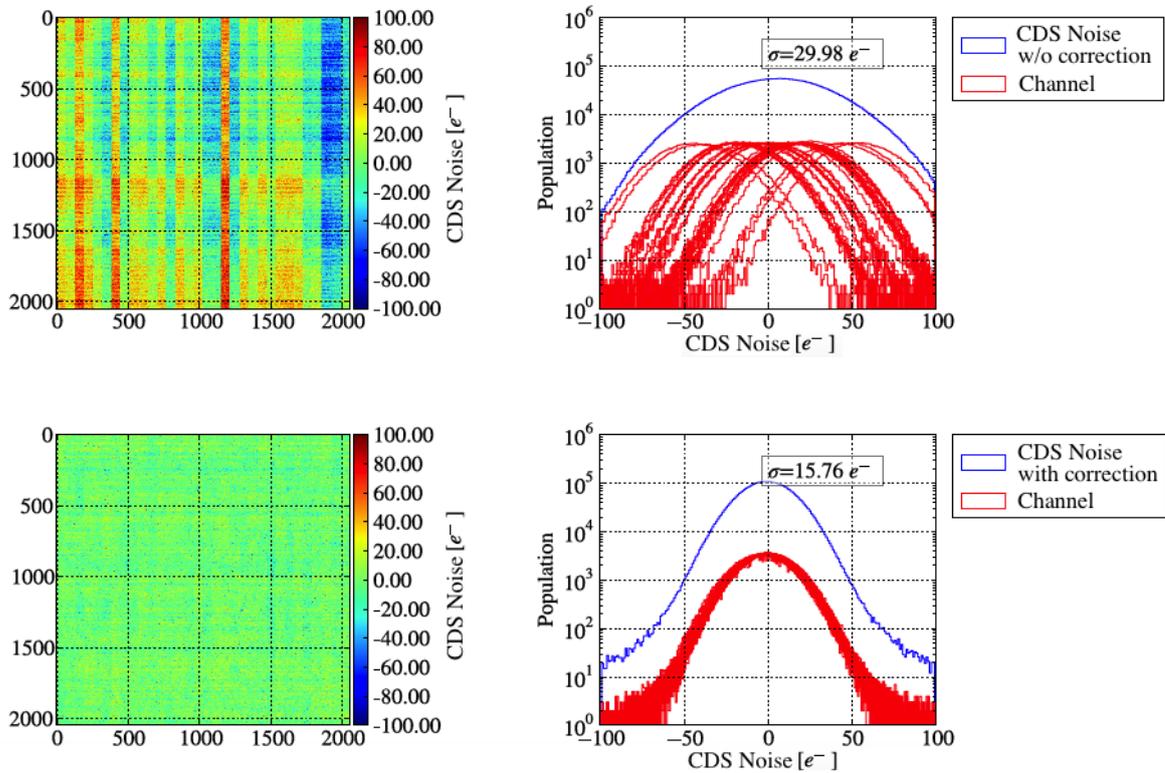


Figure 4.24: *Effect of the reference pixels correction #4 (subtraction top-bottom, then left-right) on the distribution of the CDS noise of the detector. Top: CDS noise without correction and histograms of the 32 channels of the detector. Bottom: CDS noise with correction of reference pixels and histograms of the 32 channels of the detector.*

## 4.7 Flux calculation

The estimation of the flux from the observed sources thus passes first through a measurement of the signal ramp  $f(t) = at + b$ , with  $t$  the time since the beginning of the observation,

$b$  the zero point and  $a$  the slope of the ramp (which is the flux coming from the source). Fitting a line on a linear ramp is not, a priori, a complicated exercise, but since each reading is non-destructive, the successive groups are correlated by Poisson noise with a contribution of non-negligible readout noise. For the fitting of the signal ramp, one would therefore use a fit method where the error is limited to the Poisson error. However, a complication arises from the readout noise, which is not negligible for the exposure times and the flux of the objects observed, and should be taken into account at low flux. It is also required to have a quality control of the fit, in order to know if the data are compatible with the hypothesis of a linear response. The on-board electronics uses a  $\chi^2$  minimization method calculated with the least squares method, which can be analytically computed. This classical method, described in the following section, has the advantage of providing the best possible estimate of the value of the flux we are looking for, as well as a criterion for the quality of the fit (the  $\chi^2$ ). We will use a minimization method to find the most probable value of the flux ( $F_{fit}$ ) by comparing the data (the averaged groups) with a theoretical model (in this case a line of equation  $F_{fit} \cdot t$ ). For the description of this method, we consider a MACC  $[N, M, D]$  readout mode with  $N$  the number of groups,  $M$  and  $D$  the number of images read and the number of drops, respectively. Each of the groups being averaged over  $M$  images, the associated readout error is decreased by  $\sqrt{M}$ . The signal value of each group corresponds to the average value of all the images in the group:

$$g_i = \frac{1}{M} \sum_{j=0}^M s_j \quad (4.15)$$

where  $s_j$  is the signal of an image. The associated error is:

$$\epsilon_i^2 = g_i + \frac{\sigma_{readout}^2}{M}. \quad (4.16)$$

Instead of focusing on the direct space of the  $g_i$ , where groups of mean images are correlated because the signal is integrated, we will work in the space of group differences ( $F_i = g_{i+1} - g_i$ ) in which the Poisson fluctuations are not correlated. Another advantage of this choice is that a knowledge of the ordinate at the origin of the ramp becomes unnecessary.

#### 4.7.1 Case without readout error

First, we are interested in the case where the error is dominated by the Poisson error of the signal ( $F_{fit} \gg \sigma_{readout}^2$ ). Working in the space of group differences, the  $\chi^2$  of the fit is written as follows:

$$\chi^2 = \sum_{i=1}^{N-1} \frac{(F_i - F_{fit})^2}{F_{fit}}. \quad (4.17)$$

In this equation, it can be noted that the Poisson error is taken from the fitted signal, since this is derived from the expected value of the signal. The fitted flux  $F_{fit}$  can be calculated by

zeroing the first derivative of  $\chi^2$  with respect to  $F_{fit}$ :

$$\begin{aligned}
\frac{\partial \chi^2}{\partial F_{fit}} = 0 &\Rightarrow \sum_{i=1}^{N-1} \frac{\partial}{\partial F_{fit}} \left[ \frac{(F_i - F_{fit})^2}{F_{fit}} \right] = 0 \Rightarrow \\
&\Rightarrow \sum_{i=1}^{N-1} \frac{-2(F_i - F_{fit}) F_{fit} - (F_i - F_{fit})^2}{F_{fit}^2} = 0 \Rightarrow \\
&\Rightarrow \sum_{i=1}^{N-1} (-2F_i F_{fit} + 2F_{fit}^2 - F_i^2 - F_{fit}^2 + 2F_i F_{fit}) = 0 \Rightarrow \\
&\Rightarrow \sum_{i=1}^{N-1} (F_{fit}^2 - F_i^2) = 0 \Rightarrow (N-1)F_{fit}^2 - \sum_{i=1}^{N-1} F_i^2 = 0 \Rightarrow \\
&\Rightarrow F_{fit} = \sqrt{\frac{\sum_{i=1}^{N-1} F_i^2}{N-1}}.
\end{aligned} \tag{4.18}$$

The flux error is obtained by computing the second derivative of the  $\chi^2$  with respect to  $F_{fit}$  and applying the Fisher information matrix principle for the calculation of the variance of an unbiased estimator:

$$\begin{aligned}
\sigma_{F_{fit}}^2 &= 2 \left( \frac{\partial^2 \chi^2}{\partial F_{fit}^2} \right)^{-1} = 2 \left[ \frac{\partial}{\partial F_{fit}} \left( N-1 - \frac{\sum_{i=1}^{N-1} F_i^2}{F_{fit}^2} \right) \right]^{-1} = \\
&= 2 \left( 2 \frac{\sum_{i=1}^{N-1} F_i^2}{F_{fit}^3} \right)^{-1} = \frac{F_{fit}^3}{\sum_{i=1}^{N-1} F_i^2} = \frac{\frac{F_{fit}^3}{N-1}}{\frac{1}{N-1} \sum_{i=1}^{N-1} F_i^2} = \\
&= \frac{1}{F_{fit}^2} \cdot \frac{F_{fit}^3}{N-1} = \frac{F_{fit}}{N-1} \Rightarrow \sigma_{F_{fit}} = \sqrt{\frac{F_{fit}}{N-1}}.
\end{aligned} \tag{4.19}$$

As shown in Figure 4.25 (obtained by a Monte Carlo simulation, with a readout error set to  $\sigma_{readout} = 10 \text{ e}^-$ ), this formula works at high flux ( $> 1 \text{ e}^-/\text{s/pixel}$ ). At low flux, there is a difference between the error on the flux and the statistical error, which is too large: this is explained by the presence of a significant readout noise, which becomes dominant at this flux regime. It can thus be seen that for fluxes  $< 1 \text{ e}^-/\text{s/pixel}$  (typically fluxes less intense than the science flux), corresponding to noise measurements, the readout noise cannot be neglected.

#### 4.7.2 Case with readout error

By adding the readout error to the Poisson error in the  $\chi^2$  estimation, we obtain the following equation:

$$\chi^2 = \sum_{i=1}^{N-1} \frac{(F_i - F_{fit})^2}{F_{fit} + \frac{2\sigma_{readout}^2}{M}}. \tag{4.20}$$

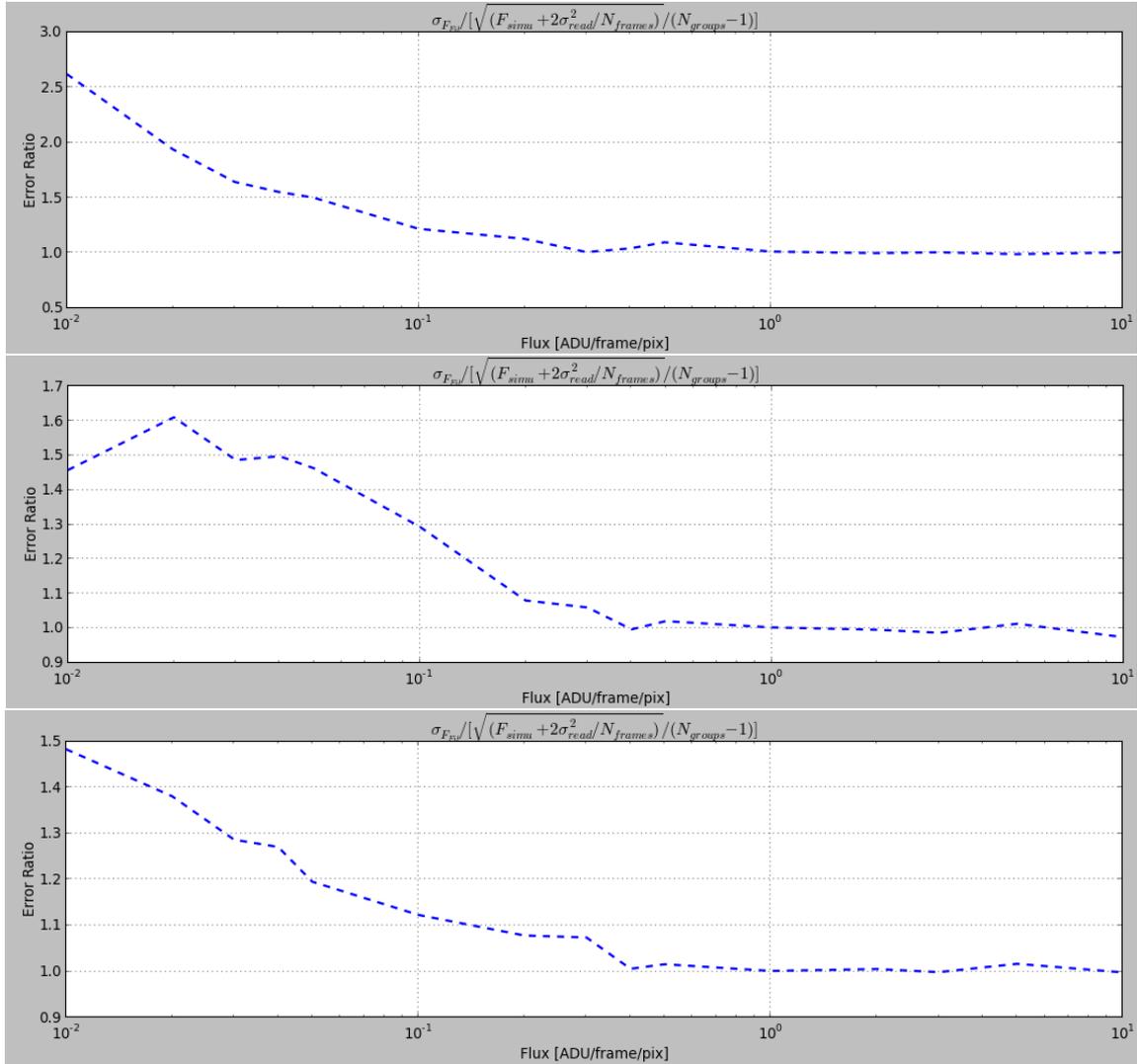


Figure 4.25: Ratio between Monte Carlo simulation statistical error and combination of Poisson and readout error for different fluxes, without correction for the readout noise on top, with correction for the readout noise in the middle and with maximum likelihood correction in the bottom. For each flux value the statistical error was computed simulating the signal of 1000 ramps, each consisting of 394 frames (15 groups of 16 frames and 11 drops, to carry out spectroscopic MACC readout).

The readout error term  $\frac{\sigma_{readout}^2}{M}$  is multiplied by 2 because we calculate image differences, so the readout error is counted twice. Let us minimize the  $\chi^2$  in order to update the expression for  $F_{fit}$ . To simplify the following calculations, we put  $F'_{fit} = F_{fit} + \frac{2\sigma_{readout}^2}{M}$ . For simplicity, the quantities  $a_i$ ,  $b$  are defined as

$$\begin{cases} a_i = F_i + \frac{2\sigma_{readout}^2}{M} \\ b = F'_{fit} \end{cases}. \quad (4.21)$$

The first derivative of  $\chi^2$  with respect to  $F'_{fit}$  thus becomes

$$\begin{aligned} \frac{\partial \chi^2}{\partial b} &= \frac{\partial}{\partial b} \left[ \sum_{i=1}^{N-1} \frac{(a_i - b)^2}{b} \right] = \sum_{i=1}^{N-1} \frac{-2(a_i - b)b - (a_i - b)^2}{b^2} = \sum_{i=1}^{N-1} (a_i - b) \frac{-2b - a_i + b}{b^2} = \\ &= - \sum_{i=1}^{N-1} (a_i - b) \frac{a_i + b}{b^2} = \sum_{i=1}^{N-1} \frac{b^2 - a_i^2}{b^2} = \sum_{i=1}^{N-1} \left( 1 - \frac{a_i^2}{b^2} \right) = N - 1 - \frac{1}{b^2} \sum_{i=1}^{N-1} a_i^2. \end{aligned} \quad (4.22)$$

By replacing  $a_i$  and  $b$  by their definition, we obtain the following expression for  $\chi^2$  minimization:

$$\begin{aligned} \frac{\partial \chi^2}{\partial F'_{fit}} &= N - 1 - \frac{1}{F_{fit}^2} \sum_{i=1}^{N-1} \left( F_i + \frac{2\sigma_{readout}^2}{M} \right)^2 = 0 \Rightarrow \\ &\Rightarrow F'_{fit} = \sqrt{\frac{\sum_{i=1}^{N-1} \left( F_i + \frac{2\sigma_{readout}^2}{M} \right)^2}{N - 1}}. \end{aligned} \quad (4.23)$$

Therefore, we deduce the expression for  $F_{fit}$ :

$$F_{fit} = \sqrt{\frac{\sum_{i=1}^{N-1} \left( F_i + \frac{2\sigma_{readout}^2}{M} \right)^2}{N - 1}} - \frac{2\sigma_{readout}^2}{M}. \quad (4.24)$$

The second derivative of  $\chi^2$  with respect to  $F'_{fit}$  is

$$\frac{\partial^2 \chi^2}{\partial b^2} = \frac{\partial}{\partial b} \left( N - 1 - \frac{1}{b^2} \sum_{i=1}^{N-1} a_i^2 \right) = \frac{2}{b^3} \sum_{i=1}^{N-1} a_i^2 \quad (4.25)$$

so that the flux error becomes

$$\begin{aligned} \sigma_{F_{fit}}^2 &= \frac{F_{fit}^3}{\sum_{i=1}^{N-1} \left( F_i + \frac{2\sigma_{readout}^2}{M} \right)^2} = \frac{F_{fit}^3}{(N - 1)F_{fit}^2} = \frac{F'_{fit}}{N - 1} \Rightarrow \\ &\Rightarrow \sigma_{F_{fit}} = \sqrt{\frac{F_{fit} + \frac{2\sigma_{readout}^2}{M}}{N - 1}}. \end{aligned} \quad (4.26)$$

Figure 4.25 shows an improvement in the estimation of the low flux error. However, a systematic overestimation of  $\chi^2$  for fluxes below 0.7 e<sup>-</sup>/s remains.

### 4.7.3 The maximum likelihood correction

In order to correct the low-flux error, the flux can be calculated using a maximum likelihood method. It is assumed that the flux value has been measured  $N$  times (the values of  $F_i$ ) and that the probability distribution of these values is Gaussian with average  $F$  and  $\sigma$  as standard deviation. The probability to obtain a value of the flux  $F_i$  at the  $i^{th}$  measure can be written as

$$P(F_i) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(F_i-F)^2}{2\sigma^2}}. \quad (4.27)$$

When we look at a set of  $N$  independent measures, the probability  $P$  to obtain such a set of results  $F_i$  is:

$$P = \prod_{i=1}^N P(F_i). \quad (4.28)$$

By replacing  $P(F_i)$  with Equation 4.27, we have

$$P = \left( \frac{1}{\sigma\sqrt{2\pi}} \right)^N \cdot e^{-\sum_{i=1}^N \frac{(F_i-F)^2}{2\sigma^2}}. \quad (4.29)$$

The most probable value of  $F$  is the one which maximizes the probability  $P(F)$ . This method for obtaining the flux is called the *maximum likelihood*. At this point we put

$$X = -2 \log P = -2N \log \left( \frac{1}{\sigma\sqrt{2\pi}} \right) + \sum_{i=1}^N \frac{(F_i - F)^2}{\sigma^2}. \quad (4.30)$$

In this case, maximizing  $P(F)$  in Equation 4.28 is equivalent to minimizing  $X$ . If at first order approximation, we assume that the error is constant, minimizing the term  $\sum_{i=1}^N \frac{(F_i-F)^2}{\sigma^2}$  by putting  $\sigma^2 = F$  and by resolving  $\frac{\partial X}{\partial F} = 0$ , we obtain again Equation 4.17. The maximum likelihood method thus gives at first approximation the same result as the  $\chi^2$  method already seen. However, the factor  $-2N \log \left( \frac{1}{\sigma\sqrt{2\pi}} \right)$  was not included in the previously seen  $\chi^2$ , where the low flux error is important. If we now take the exact  $\chi^2$  formula with readout error, which corresponds to Equation 4.20, and adopt the same approach by taking  $-2 \log P$ , we obtain a new  $\chi^2$ :

$$\chi^2 = \sum_{i=1}^{N-1} \frac{(F_i - F_{fit})^2}{F_{fit} + \frac{2\sigma_{readout}^2}{M}} + (N-1) \log \left( F_{fit} + \frac{2\sigma_{readout}^2}{M} \right). \quad (4.31)$$

With the same minimization method and the same substitutions defined in Equation 4.21, we can obtain  $F_{fit}$  by nullifying  $\chi^2$  first derivative:

$$\begin{aligned} \frac{\partial \chi^2}{\partial b} &= N-1 - \frac{1}{b^2} \sum_{i=1}^{N-1} a_i^2 + \frac{N-1}{b} = 0 \Rightarrow b^2 + b - \sum_{i=1}^{N-1} \frac{a_i^2}{N-1} = 0 \Rightarrow \\ &\Rightarrow b = \frac{-1 \pm \sqrt{1 + 4 \sum_{i=1}^{N-1} \frac{a_i^2}{N-1}}}{2} \Rightarrow b > 0 \Rightarrow b = \frac{\sqrt{1 + 4 \sum_{i=1}^{N-1} \frac{a_i^2}{N-1}} - 1}{2}. \end{aligned} \quad (4.32)$$

This means that  $F_{fit}$  results to be

$$F_{fit} = \frac{\sqrt{1 + \frac{4}{N-1} \sum_{i=1}^{N-1} \left( F_i + \frac{2\sigma_{readout}^2}{M} \right)^2} - 1}{2} - \frac{2\sigma_{readout}^2}{M}. \quad (4.33)$$

We have thus determined an exact analytical formula which gives the best possible determination of the flux, taking into account the Poisson and readout errors (see Figure 4.25). This formula is simple enough to be used by on-board electronics to calculate the flux in flight, and in such a way that it is not necessary to return all the groups of the ramp.

#### 4.7.4 Conclusions on flux calculation

It is possible to program the SCE to return to the NI-WE only the  $M$  images of the  $N$  groups. For an acquisition of the MACC [15, 16, 11] spectroscopic type, the amount of data received consists in  $15 \times 16$  images to be processed by the NI-DPU (size divided by a factor  $\approx 2$  with respect to an UTR). The NI-DPU will first construct the average image for each group (*coadding*, done by the DCU) and then these averaged images will be processed with the previous minimization method to extract:

- the flux received by the pixel,  $F_{fit}$  (Equation 4.33);
- the quality criterion for  $F_{fit}$ ,  $\chi_{Euclid}^2$  (Equation 4.31).

This processing will be carried out in flight, and for each exposure only these two pieces of information will be sent to the ground for each pixel, i.e. only one matrix of 16-bit words for the flux and one of 8-bit words for  $\chi_{Euclid}^2$ . The sequence of acquisition/processing is shown in Figure 4.26. Note that despite the reduction to only 2 “images” (that of the slope and that of the  $\chi_{Euclid}^2$ ) representing about 25 GB/day, this data will be compressed to gain a factor of 2 on the data volume, in order to save memory and provide the minimum possible amount of telemetry.

## 4.8 NISP detectors characterization

The H2RG detectors introduced in this chapter are tested and delivered to the Euclid consortium by NASA. 20 detectors will be selected as “flight” detectors and delivered to the *Centre de Physique des Particules de Marseille* (CPPM) to be characterized and integrated into the focal plane of the NISP instrument. The instrument will then be delivered to ESA in 2019. Here, we describe the detector selection specifications for the Euclid mission as well as the operational constraints in flight. The selection criteria for the detectors are summarized in Table 4.3. It is required for the 95% of the pixels (except for the dark current) to be “operable” (usable for science). The criteria include quantities already defined such as dark current, quantum efficiency, or detector readout noise. Other selection criteria complement this list, but we will focus here on the aspects where a pixel-by-pixel characterization is provided. These selection criteria are set in order to fulfill the scientific objectives of the NISP instrument. The characterization of Euclid H2RG detectors will be carried out before the launch. We will

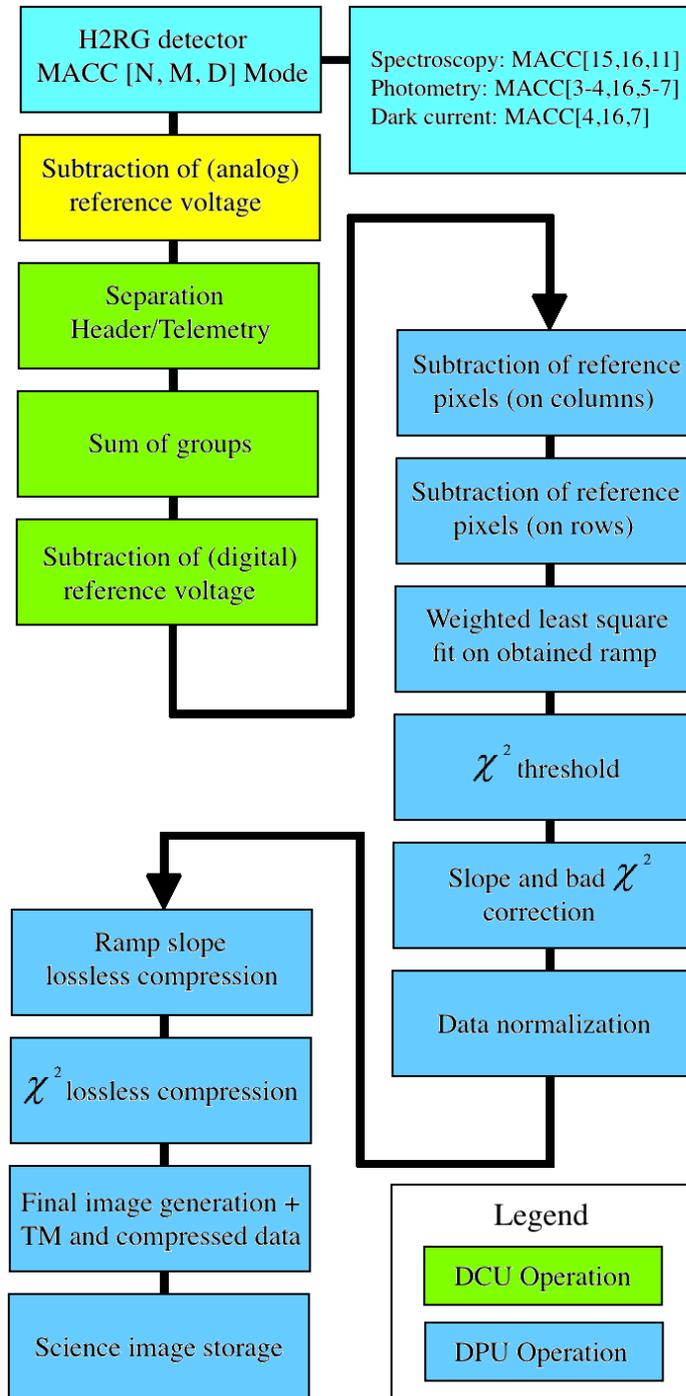


Figure 4.26: Representation of data processing by NISP on-board electronics [60].

Parameters	Specifications (95% of pixels)
Total noise	$\leq 13 e^-$ for spectroscopic mode $\leq 9 e^-$ for photometric mode
Dark current	$< 0.07 e^-/s$
Non-linearity	5% from 6 to 60 $ke^-$
Quantum efficiency	75% between $[1.1, 2] \mu m$

Table 4.3: Criteria for the selection of Euclid detectors by NASA.

first describe the data processing steps and the sensor characterization requirements. Then we will describe the test and analysis methods developed at the CPPM, the expected performance and the results of detector characterization available so far.

## 4.9 Data processing on ground and characterization objectives

Before using the information about the flux received by the pixels, it is necessary to calibrate their responses. This involves a number of steps that we will describe and which are summarized in Figure 4.27:

- isolate bad pixels from a predefined map;
- isolate pixels hit by cosmic rays;
- subtract the dark current;
- convert the received flux ( $ADU/frame^2$ ) to  $e^-/s$ ;
- correct non-linear response effects;
- correct impact of neighboring pixels;
- correct persistence effects.

Calibration procedures often rely on predefined procedures that depend on the *a priori* knowledge of the individual properties of the pixels. This knowledge requires a fine characterization of all the pixels, i.e. a long data acquisition to carry out the previous corrections. The corrections for steps 1 and 3 to 7 depend on the intrinsic properties of the detector and require dedicated maps. In addition, steps 2 and 7 depend on the observation strategy: the former requires a rejection algorithm, the latter a knowledge of the persistence response of the pixels in the detector. In the framework of Euclid, in-flight calibrations will be limited. This requires, on one hand, fine characterizations performed before the launch, and on the other hand, the adoption of procedures to be applied to the data. We will now detail the needs for some of these procedures.

---

<sup>2</sup>A frame is the time it takes for the detector to acquire a single image, i.e. 1.41 s.

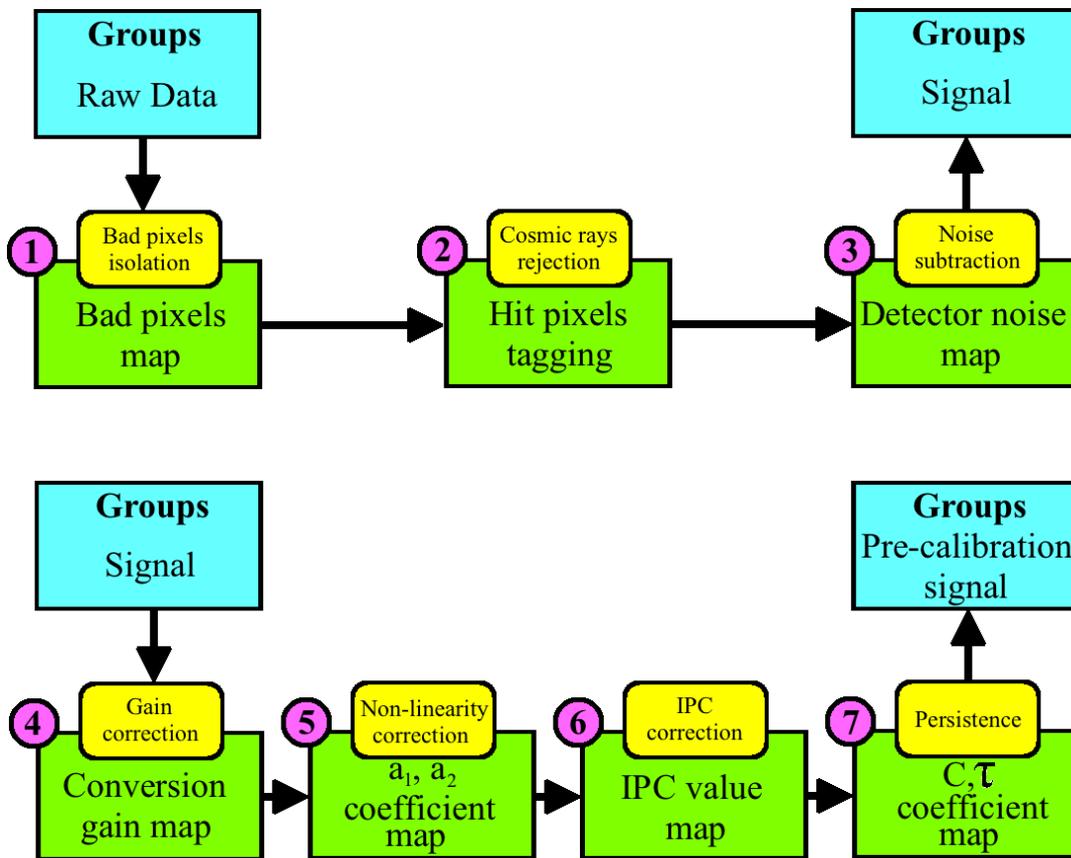


Figure 4.27: Diagram showing raw data pre-processing on ground.

## 4.10 Requirements for the characterization

The characterization of flight sensors must provide a mapping of the parameters defining the response of the pixels in order to respond to raw data processing requirements. Characterization maps are divided into two categories, those that require flat field illumination and those that require no light. The main parameters that are required to be known are the total noise of the detector (related to its readout mode), non-linearity, conversion gain, inter-pixel capacitance and persistence. Other parameters are taken into account for the determination of pixel operability, but these will not be dealt with. In this section we will describe the data required for characterization, as well as the instrumental requirements that this imposes.

### 4.10.1 Detector noise correction

As we saw in the previous chapter, the photometric and spectroscopic modes of observation require a MACC readout, that is to say a continuous ramp of averaged groups. The noise measured using these modes always contains the two components of readout and dark current

Poisson noise, which form the total noise. For the characterization, the goal is to be able to dissociate at pixel level the parameters influencing the total noise, that is to say the readout noise and the stochastic noise originating from the dark current [56]. Figure 4.28 illustrates the different noise regimes of the detector as a function of the incident flux for a UTR exposure of 5000 images with a readout noise of  $10 e^-$ . The blue curve represents the contribution of readout noise ( $\sigma_{readout}$ ), the green one shows the contribution of Poisson noise and the red one is the quadratic sum of the two. One can see two noise regimes, the first, below a flux of  $2 \times 10^{-2} e^-/s$ , is dominated by the readout noise. Above this flux, the total noise is dominated by Poisson contribution. This measurement of noise is associated with the measurement of the dark current, which is the result of two independent processes: an internal process of spontaneous generation of charges in the pixel [61], and an external process resulting from the arrival of “undesirable” photons on the pixel. These unwanted photons are related to the environment of the detector and also called *stray light*. The measurement of the dark current is necessary in order to subtract this contribution to the science signal (third correction on the diagram in Figure 4.27). The purpose of the characterization is to obtain measurements of the dark current of the order of  $10^{-3} e^-/s$  with an accuracy of less than  $4 \times 10^{-3} e^-/s$  (Requirement NISP-C-1012) for operating temperatures between 85 and 100 K. Measurements at these levels of accuracy require a control of the external and internal variables influencing the dark current. For internal processes, the dominant component is the operating temperature. By imposing a constraint on the operating temperature, which must be stable at mK order, it is then possible to avoid thermal variations generating noise. Concerning the external contributions to the dark current, it is necessary to minimize light leakages.

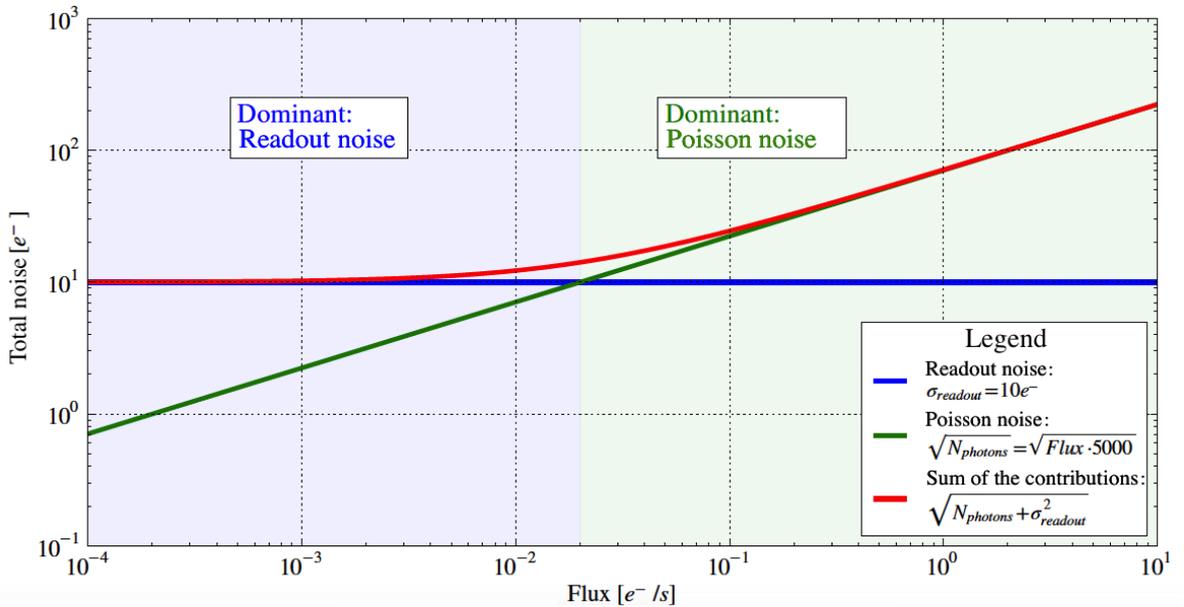


Figure 4.28: Contributions of the two noise components as a function of the flux for a 5000 frames exposure.

### 4.10.2 Conversion gain

The detector will generate charge carriers ( $e^-$ , holes) and, once digitized, the signal is in ADU. In order to obtain the signal in number of charge carriers, it is necessary to multiply the ADU signal by the conversion gain  $G_{conv}$  (Equation 4.6):

$$S_{carriers} = G_{conv} \times S_{ADU}. \quad (4.34)$$

In order to determine the conversion gain of the detection system, the *Photon Transfer Curve* (PTC) method is used [62]. This method consists in evaluating the relationship between the measured photon flux value and its variance. For a given flux, the signal for a pixel of the detector is written as

$$S_{ADU} = \frac{N_{carriers}}{G_{conv}} + \sigma_{readout} \quad (4.35)$$

with  $S_{ADU}$  the signal of the pixel expressed in ADU,  $N_{carriers}$  the number of carriers generated in the pixel by an incoming photon flux and  $\sigma_{readout}$  the readout error on the signal. The error associated with this  $S_{ADU}$  signal can be written as follows:

$$\sigma_{S_{ADU}}^2 = \frac{\sigma_{N_{carriers}}^2}{G_{conv}^2} + \sigma_{readout}^2. \quad (4.36)$$

Since the number of carriers generated is related to the number of photons received by the detector, the  $N_{carriers}$  variable follows a Poisson distribution, so that the variance is equal to the mean of the distribution:

$$\sigma_{N_{carriers}}^2 = \langle S_{carriers} \rangle = G_{conv} \cdot \langle S_{ADU} \rangle. \quad (4.37)$$

By replacing in Equation 4.36 the value of  $\sigma_{N_{carriers}}^2$  obtained with Equation 4.37, one gets

$$\sigma_{S_{ADU}}^2 = \frac{\langle S_{ADU} \rangle}{G_{conv}} + \sigma_{readout}^2. \quad (4.38)$$

This shows that the variance and the mean value of the signal are linked by a linear equation such that if we plot the relation  $\sigma_{S_{ADU}}^2 = f(\langle S_{ADU} \rangle)$  we obtain a line with slope  $\frac{1}{G_{conv}}$  (see Figure 4.29). This method requires obtaining the response of the pixels for illumination from 0.5 to 64  $e^-/s$ , which is the range of flux arriving on the detectors in science mode. In order to retrieve the signal in number of photons, one must then divide by the quantum efficiency of the detector ( $\eta$ ):

$$S_{photons} = \frac{S_{carriers}}{\eta}. \quad (4.39)$$

In order to measure the relative quantum efficiency on the detection system with a precision of more than 5%, the detector requires a uniform illumination profile (better than 1%) on the detector surface, so that it is necessary to provide monochromatic illumination at wavelengths between 0.9 and 2.0  $\mu\text{m}$  in 50 nm steps.

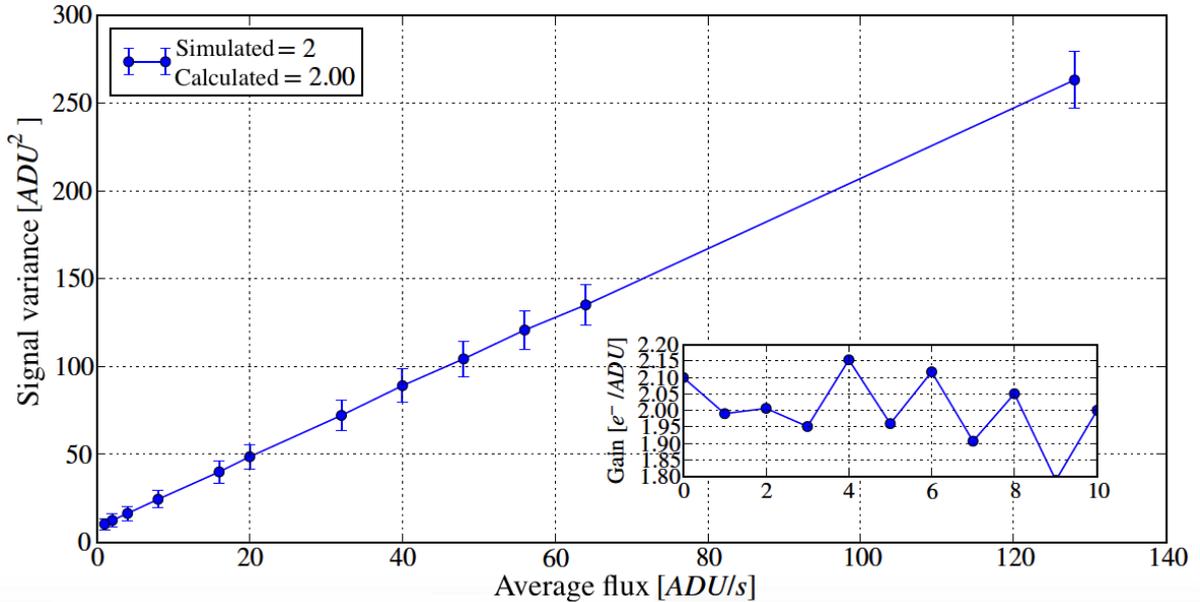


Figure 4.29: Example of a PTC obtained from a simulation with a gain  $G_{conv} = 0.5 \text{ e}^-/\text{ADU}$ , readout noise  $\sigma_{readout} = 10 \text{ e}^-$  and fluxes from 1 to 128  $\text{e}^-/\text{s}$ .

### 4.10.3 Inter-pixel capacitance

Concerning the signal of a pixel, there is an electrostatic potential effect that can be modeled by a capacitance between neighboring pixels (explained in §4.3.4). Figure 4.30 is a representation of the effect of IPC on the image of a detector, assuming a parasitic capacitance only on the nearest neighbors. We see the appearance of a characteristic IPC pattern in the form of a cross that shows the impact of such signals on a science image. Several methods can be used for the measurement of IPC. The first uses a point source (at the pixel scale) with the help of a radioactive X-ray source of  $^{55}\text{Fe}$  or a laser [51]. The second method consists in changing (that is, increasing) the  $V_{reset}$  value of a pixel of the detector (so as to introduce an artificial signal [51]). By generating an important signal on a pixel, it is thus possible to measure the capacitance effect between pixels; however, it is then necessary to use a technique capable of iterating this measurement on the entire pixel matrix of the detector. This second method is the one which will be adopted for Euclid detectors, with a firmware allowing the creation of a grid of 1 to N pixels with a different  $V_{reset}$ , thus creating a grid of pixels with a strong signal. On the image produced, the IPC can be measured by reconstructing the induced signal on the near neighbors of a pixel. This pattern is then shifted and repeated on the other pixels to cover the whole matrix (necessary for the sixth correction in Figure 4.27).

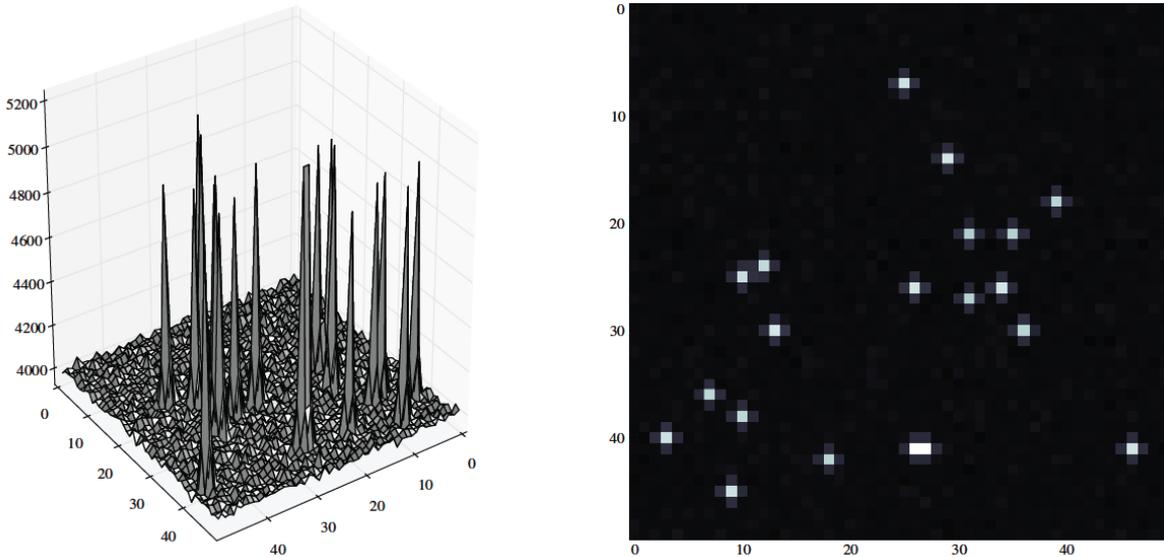


Figure 4.30: *Simulation of the effect of IPC on a detector for point-like impacts.*

#### 4.10.4 Non-linearity

The non-linear response of the pixels of the detector implies that the flux is systematically underestimated if it is not corrected. This non-linearity correction must be able to reconstruct the linear response to better than 1%, which is achievable by approximating the response of the pixel as a polynomial of the following type [63]:

$$S_{measured}(t) = S_{real}^2(t) \cdot a_2 + S_{real}(t) \cdot a_1 + a_0 \quad (4.40)$$

with  $a_1$  corresponding to the angular coefficient of the linear response of the pixel (and therefore the conversion gain),  $S_{measured}$  the signal measured in ADU and  $S_{real}$  the corrected signal in ADU (or  $e^-$  if the conversion gain is applied). These coefficients vary from one pixel to another and, according to the readout modes used, the characterization must therefore be able to provide, for each of the spectroscopic and photometric modes, the map of coefficients  $a_0$ ,  $a_1$  and  $a_2$ , necessary for this correction per pixel (the fifth correction in Figure 4.27). The required measurements are long and complex because the detector has to be explored throughout the entire operation range of flux values. Flat field illumination of the detector is required together with a variable flux in order to check the agreement between the low-flux and high-flux non-linearity coefficients.

#### 4.10.5 Persistence

The persistence signal is a phenomenon of charge trapping during exposure: the trapping of charges during subsequent exposures in the sensitive surface of the detector induces a measurable remanent image for the further acquisitions, detrimental for science performances. The

measurement of the persistence signal requires knowledge of the pixel response to the incident flux and the time constants of the non-volatile image produced, in order to obtain usable characterization maps. For this purpose, it is necessary to alternate between a uniform illumination and a measurement of dark current, in order to observe the behaviour of the non-volatile image.

#### 4.10.6 Summary of requirements for the characterization

Table 4.4 summarizes the needs for each step of the characterization, in order to fulfill data pre-processing on ground and associated instrumental requirements. On the basis of these requirements, the specifications for the detector characterization can be obtained. The characterization of the NISP instrument detectors is done at the CPPM, while the acceptance and selection tests are conducted by NASA. During the characterization period, 20 H2RG detectors will be delivered to the laboratory. These detectors will follow the characterization plan illustrated in Figure 4.31, which includes: an integration period (reception, testing and cold-setting) of 5 days, a characterization period with a forecast of 24 days per detector, and finally a reset to room temperature and a 4-day storage procedure. Once the characterization is complete, 16 detectors will be integrated on the focal plane of the instrument and then the entire focal plane will be mounted in the NISP instrument, which is to be delivered to ESA by 2019.

Parameter	Specifications	Testbed
Dark current in UTR mode	$\approx 10^{-3} \text{ e}^-/\text{s}$	No light leakage, stable temperature
Conversion gain	Accuracy on relative $\eta < 5\%$	Monochromatic illumination, flat field at 1%, flux between 1 and $64 \text{ e}^-/\text{s}$
Non-linearity	Correction $< 1\%$	Flat field illumination
Inter-pixel capacitance	IPC $< 0.3\%$	Reset pixels per grid
Persistence	Illumination then dark on several flux values	Synchronization between illumination and acquisition

Table 4.4: Summary of needs for characterization and instrumental requirements.

## 4.11 Characterization setup

The sensors of the NISP instrument shall be characterized at operating temperatures from 80 K to 100 K, not exceeding temperature gradients greater than 1 K/minute during the warm-up and reheating steps. In order to characterize all the detectors at low temperature, two cryostats, *Pégase* and *Andromède*, have been designed, each able to accommodate two detectors. Each one of these cryostats must be able to satisfy the requirements for the characterization parameters: total noise, dark current, conversion gain, inter-pixel capacitance and persistence.

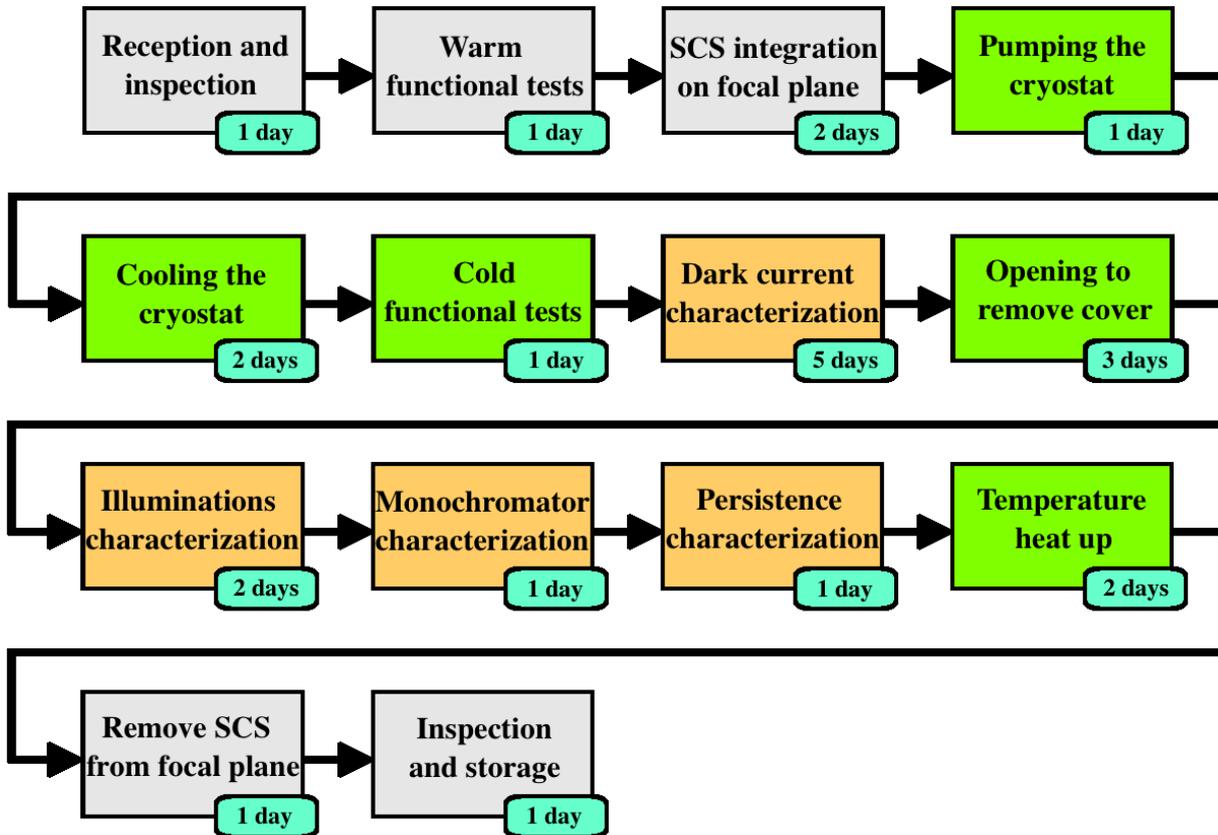


Figure 4.31: *Workflow for the characterization of Euclid SCS. In orange the periods dedicated to the characterization are represented, while the periods of tests and of environment preparation are indicated in green. The whole procedure consists in a total of 24 days allocated per detector.*

#### 4.11.1 The environment

The handling of the flight detectors imposes certain working conditions, in particular for their safety. The main sources of danger for flight detectors are:

- physical shocks damaging the integrity of the elements (SCA, SCE or CFC);
- contamination with volatile particles or molecules present in the test environment;
- electrostatic discharges during handling.

This is why the environment for handling detectors consists of an ISO 5 hood, located in an ISO 7 clean room. All present personnel are assigned an anti-static wrist strap that must be connected to ground before any manipulation of electronics. The manipulation of each of the elements constituting the SCS requires the implementation of dedicated procedures for each of the integration and extraction steps.

### 4.11.2 The *Pégase* and *Andromède* cryostats

Cryostats are enclosures in which a vacuum is created that are equipped with a cooling system. When a cryostat is cold, gaseous elements present inside tend to condense and, in the case of water, form ice crystals that can damage the electronics that will be used. If the cryostat is not under vacuum, there will also be the existence of heat exchanges by convection due to the presence of gas. The use of a pumping system to operate under vacuum allows to avoid these two problems and to prevent possible contamination of the flight detectors by volatile elements. The solution adopted is a secondary vacuum at  $\approx 10^{-6}$  mbar, which requires the use of a primary pump coupled with a turbomolecular pump. The secondary vacuum of the cryostat enclosure is provided by an Edwards pumping unit, which allows to reach voids of the order of  $10^{-6}$  mbar in 4 hours of pumping: a pressure descent of the cryostat to vacuum is shown in Figure 4.32. There are then 2 main solutions for the cooling system of the cryostats.

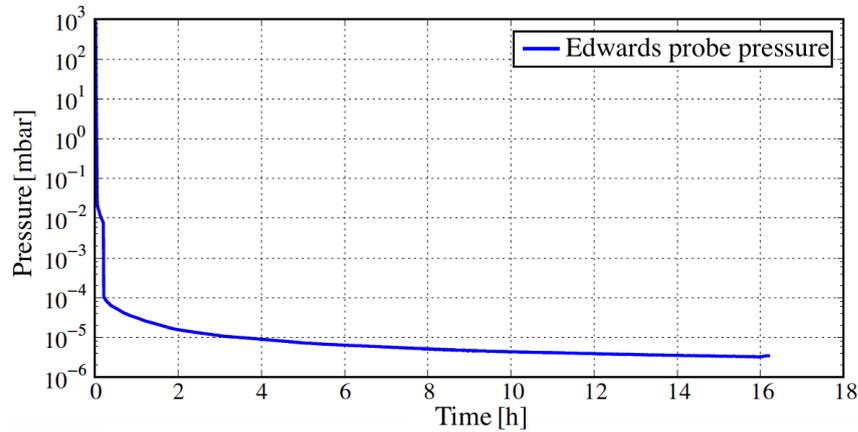


Figure 4.32: *Pégase* pressure descent to vacuum with Edwards pumping group.

The first solution consists in the use of cryofluids (liquid nitrogen at 77 K and liquid helium at 4 K). The major disadvantages in this case are considerable mass and volume of occupancy, accompanied by a constant need for availability of these cryogenic fluids. The other solution uses cryo-refrigerators, which can reach very low temperatures (below liquid nitrogen). This second, less hazardous and more practical solution is used for characterization cryostats. The cryostat is cold-pressed by the use of a Cryomech PT90 closed-loop helium-cryoprobe with a cooling capacity of 80 W at 80 K to achieve temperatures from 80 to 100 K for the focal plane<sup>3</sup>. The temperature of the other elements (sphere, baffle, screen) must not exceed 150 K in order to minimize the emission of IR photons. A thermal simulation of the cryostat (Figure 4.34) with a minimum temperature of 80 K on the focal plane (top left) and maximum temperatures  $< 110$  K on the thermal screen (bottom), has been used to validate the design of the cryostats. Figure 4.33 is the result of a Computer-Aided Design (CAD) of the enclosure of one of the characterization cryostats, *Pégase*, where we can see the Cryomech PT90 cryo-

<sup>3</sup>The SCE provides  $< 100$  mW power in 100 kHz readout frequency, 32 channels mode.

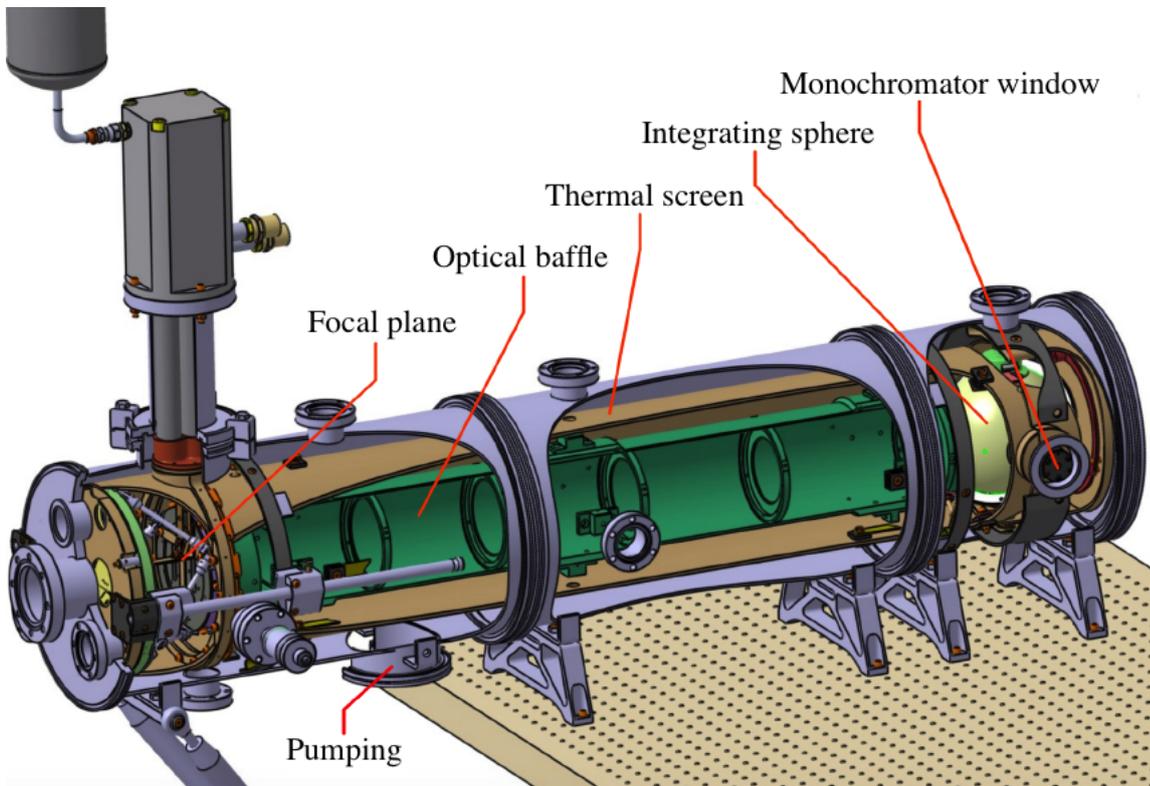


Figure 4.33: *Computer-Aided Design of the Pégase cryostat, with the main subsystems represented (focal plane with two detectors, ferrule for the pumping group, optical baffle and integrating sphere for illumination, thermal screen to protect from radiation transfer and input window for the monochromator).*

refrigerator whose cool head is located above the cryostat. There is also a ferrule below the cryostat, leading to the Edwards pumping group (located outside the clean room).

### 4.11.3 The setup for characterization in cryostats

We will describe here the different elements relevant for the characterization in cryostats and we will explain the different choices adopted.

#### Focal plane

The focal plane accommodates the detectors and consists of a machined molybdenum plate (in order to have the same Charge Transfer Efficiency as the SCA pedestal) to accommodate two SCAs (Figure 4.36 on the left). Two bushings in the plate allow the CFC cable to reach the SCEs (Figure 4.36 on the right). The stability of the focal plane temperature is ensured by the use of heaters, with 40 W power in total and located under the focal plane, and of a Proportional

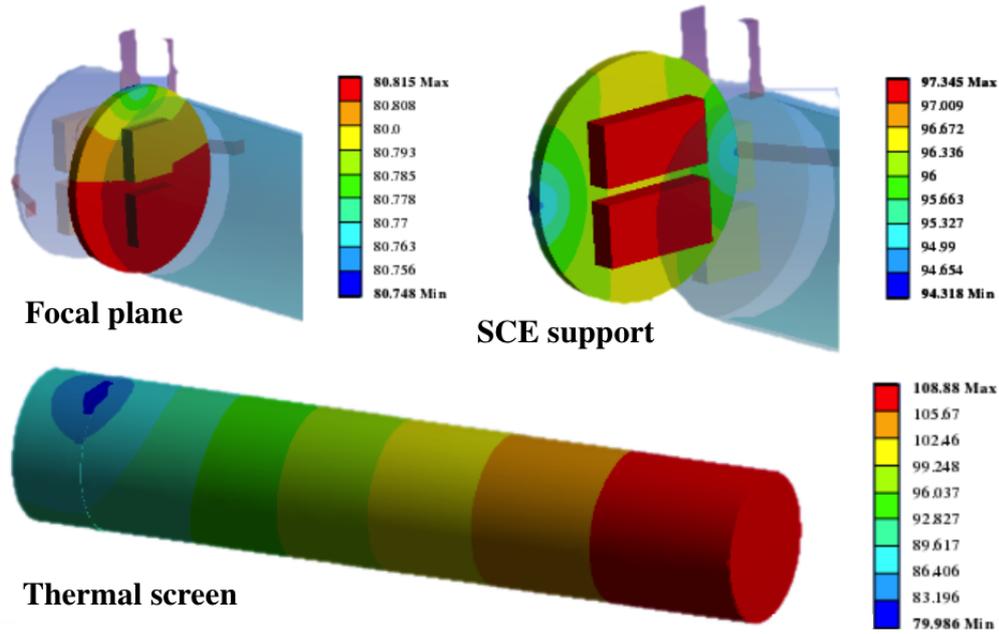


Figure 4.34: *Thermal simulation of the temperatures reached on the SCA, SCE and thermal screen support with a cold point providing 80 W of cooling power.*

Integral Derivative controller. This controller allows to ensure mK stability on the focal plane for measurements with detectors. The temperature drop with mK stabilization is illustrated in Figure 4.37, with stability of the order of mK reached at 20 h after cold-setting (the mK stability is limited by the green band on the minor plot). In order to measure dark currents of the order of  $10^{-3} \text{ e}^-/\text{s}$ , the detectors must be isolated from the sources of infrared photons which are likely to contaminate the measurement. For this purpose, all the surfaces which are not used for the illumination system will be covered with black coating effectively absorbing the parasitic infrared photons. This infrared-absorbing black layer is Acktar's Fractal Black coating [64] conforming to ESA-issued degassing specifications, with an absorbance  $\alpha \simeq 0.96$  and with a 5-14  $\mu\text{m}$  layer on a substrate in aluminium.

### Illumination system

In order to obtain a flat wavefront with the illumination system, an optical system composed of a spectralon<sup>4</sup> integrating sphere was chosen; this sphere, shown in Figure 4.38, has a diameter of 15 cm and is coupled to an optical baffle. The integrating sphere has three interfaces: the first, aligned with the cryostat window (shown in Figure 4.33), allows the light beam to enter, being modulated by an output slot from an external monochromator (Oriel Cornerstone 260) usable in a range  $\Delta\lambda = [0.9, 2.8] \mu\text{m}$  and controlled by LabView. The second aperture allows to place a reference photodiode (allowing to control the flux levels in the sphere) as well as

<sup>4</sup> Material developed by Labsphere with very high diffuse reflection (UV, NIR and visible).

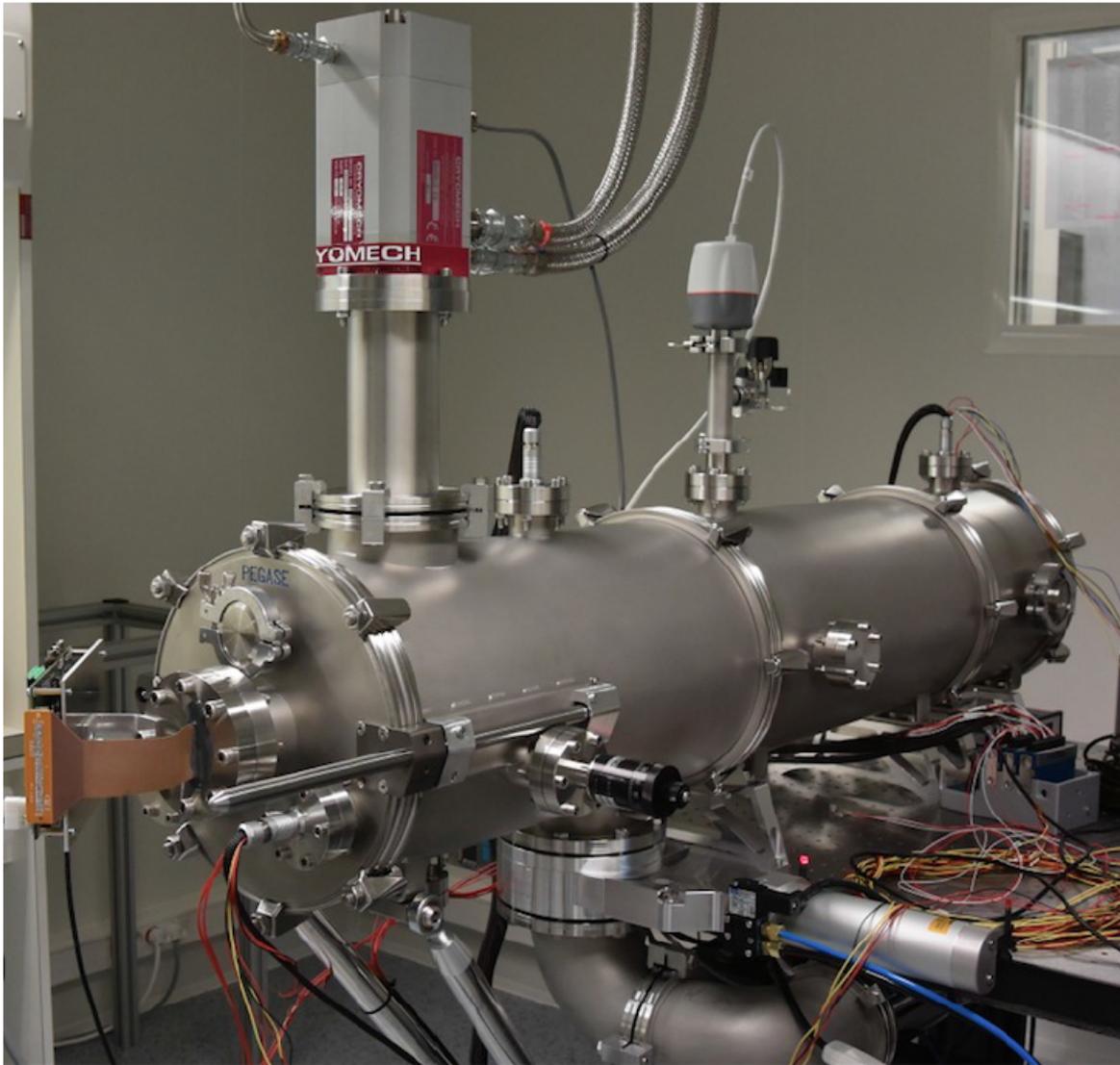


Figure 4.35: A picture of the real Pégase cryostat in the CPPM ISO 7 clean room.

LEDs (Thorlabs 1600P). The latter is the aperture letting the beam out towards the focal plane. This aperture is coupled to a 1 m optical baffle, covered with a layer similar to Acktar's coating to reduce spurious reflections and thus to achieve a precision of the order of 1 % for the uniformity of illumination on the focal plane. The performances achieved with the first *Pégase* cryostat characterization are summarized in Table 4.5.

---

<sup>5</sup>1% over large areas of the detector but some spatial variations at 2%.

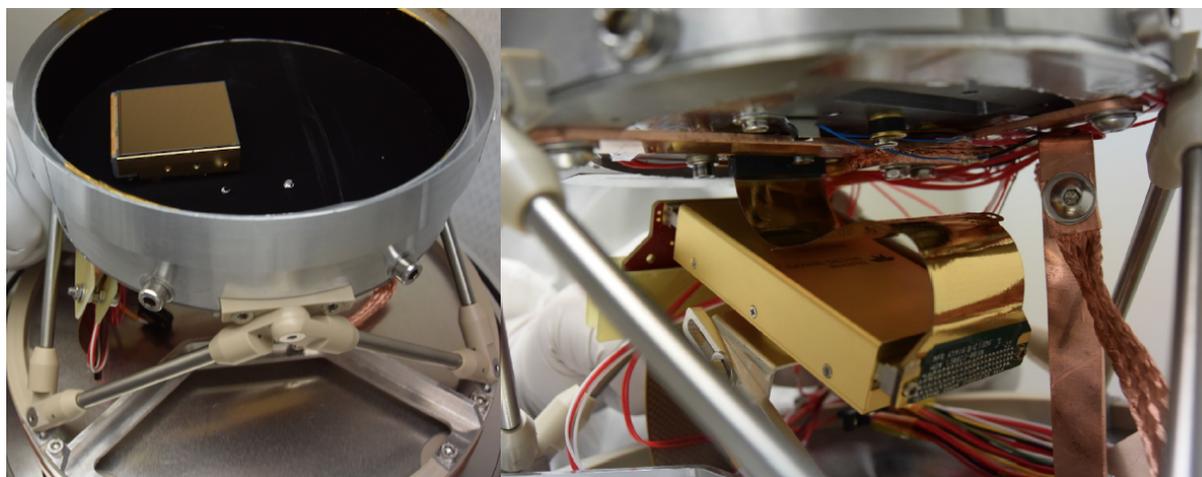


Figure 4.36: Images of the focal plane of the *Pégase* cryostat. Left: top view with blackened focal plane with SCA installed. Right: side view with the CFC cable connecting the SCA from the focal plane to the SCE installed at the rear.

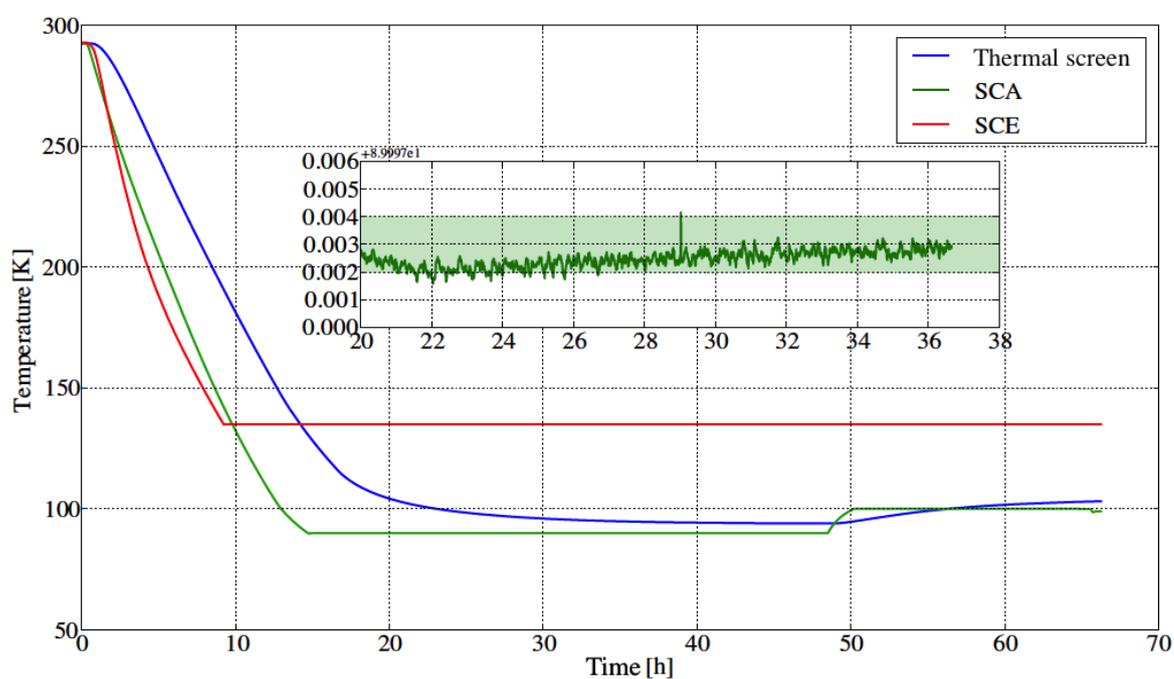


Figure 4.37: Temperature drop of the *Pégase* cryostat with the curves corresponding to the SCA, SCE and thermal screen.

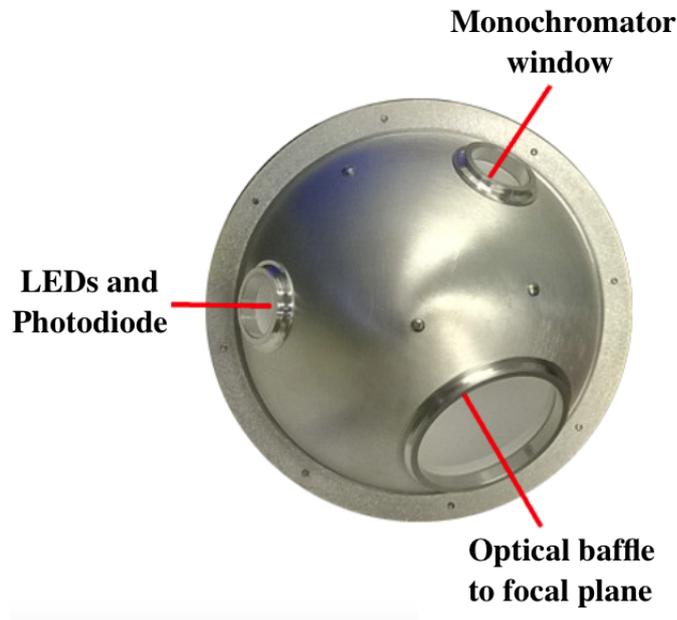


Figure 4.38: *Photo of the integrating sphere used for characterization cryostats with its three interfaces.*

## Softwares

Since acquisitions last several days, it is necessary to have a fully automated test bench capable of managing the acquisitions from the detector with a dedicated acquisition PC and to control the environment in which the detector is located. This problem is solved by the use of a set of softwares communicating with each other and thus allowing complete automation of the system. The goal is to allow the operator in charge of acquisition and analysis to interact only with an interface to launch acquisitions and with the database to launch analyses. Figure 4.39 summarizes the different interactions between each of the elements constituting the automated acquisition chain:

- *environment control*: ensured by an interface developed with LabView called *Slow Control* (SC); this interface allows to control pressure, temperature or illumination of the devices and also manages the actions in case of alarm;
- *acquisition software*: acquisitions and environment control are supported by an automated *Data Acquisition System* (DAS); this DAS, written in Python and developed by the IPNL (*Institut de Physique Nucléaire de Lyon*) team, allows the synchronization of detection chain (SCS) acquisitions with the control of the environment; the acquisitions will all be in UTR mode, which allows to analytically reconstruct all the modes, but takes up a lot of data volume; all the information (detector and environment) is recorded in HDF5 format;

Parameter	Specifications	Accomplishment
Pressure	$10^{-6}$ mbar	$10^{-7}$ mbar
$dT_{SCA}/dt$	$< 1$ K/min	$> -0.35$ K/min for cool-down $< 0.65$ K/min for heat-up
$T_{SCA}^{min}$	$< 85$ K	$\approx 70$ K
$T_{SCA}$ stability	$< 1$ mK	$< 1$ mK
Uniformity	$< 1\%$	$< 2\%^5$

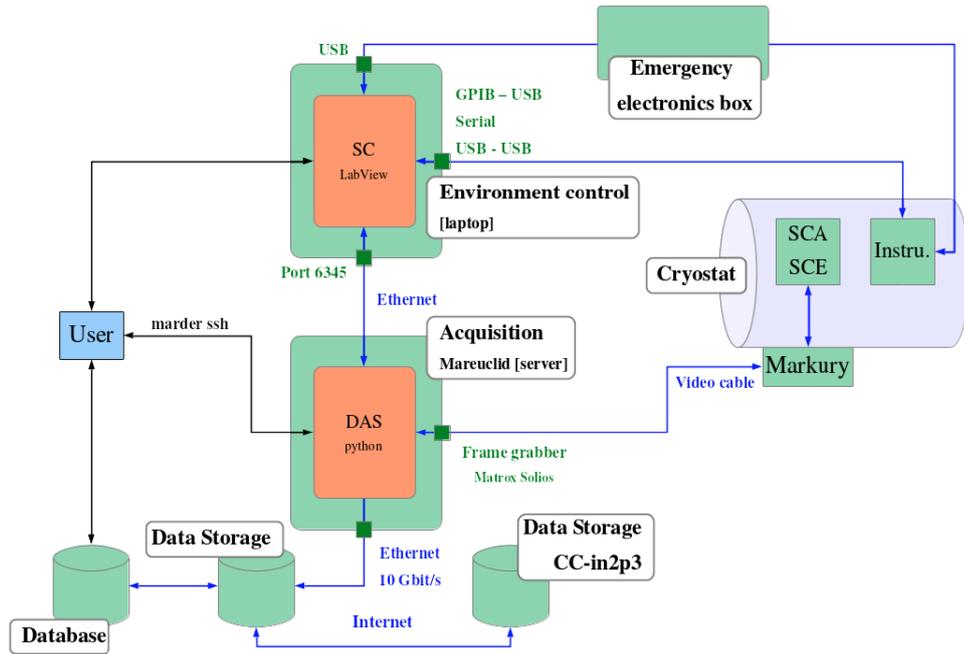
Table 4.5: Summary of performances achieved with the *Pégase* cryostat.

Figure 4.39: Diagram of interactions between the different elements of the acquisition system.

- *database*: developed in Postgre, which allows the user to easily access files resulting from acquisitions, the automation of this database allows to classify the acquisitions according to defined parameters, such as illumination, operating temperature, etc.;
- *analysis software*: allows to produce maps of detector characterization from raw data;
- *visualization software*: developed in Python, this software was used to allow visualization of data in the different file formats used by the collaboration (FITS and HDF5) and to carry out rapid analyses (Fowler-M, CDS, power spectrum, etc.).

Thus, for each acquisition, we define what we call a *sequence* (see Figure 4.40), a data set

which regroups the information about the detector configuration (polarization, gain, readout mode) and its environment (temperature, illumination). A *run* is then defined by a succession of acquisition sequences. The run is defined before the acquisition and allows to regroup all the sequences without the user having to manually launch each of them one by one. The possibility to define and command the execution of a run is provided by the *Markury electronics* module shown in Figure 4.39: this electronics can acquire SCE data from up to 32 detectors at the same time in asynchronous way.

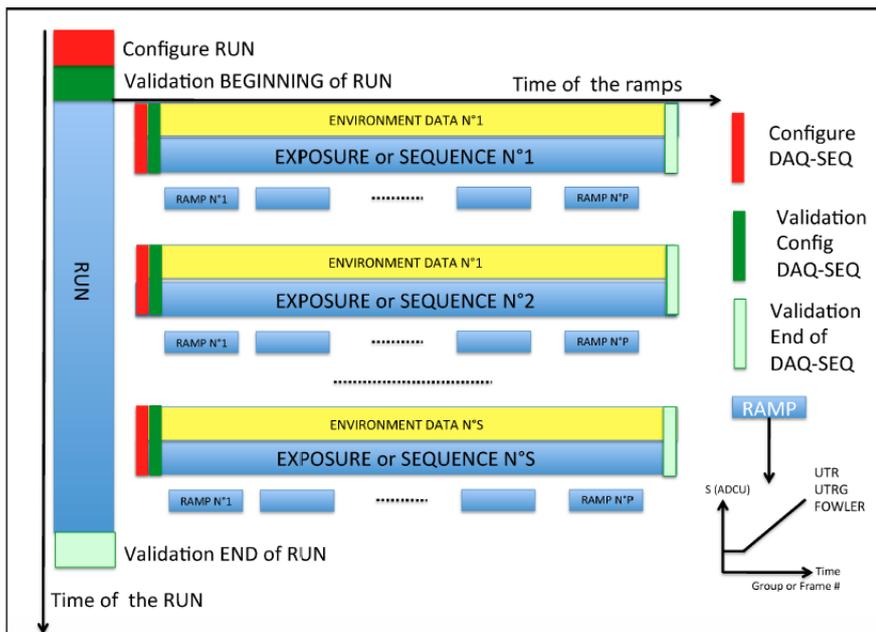


Figure 4.40: Configuration of an acquisition consisting of a succession of sequences.

## 4.12 Characterization data monitoring and analysis

The DAS provides a virtual environment interacting with several subsystems (see Figure 4.39); it interacts with:

- the SCS for data acquisition;
- the environment control, recording the data which are sent by the SC and managing the instruments;
- the Unix servers (*mareuclid* for *Andromède* or *mareuclid1* for *Pégase*) for saving data and for the so-called *quality check* (QC);
- the user (see Figure 4.41).

The DAS is installed on both available Unix servers, whose operating system is *Debian*. Each time a user wants to make an acquisition, he has to connect via *ssh* (*Secure Shell protocol*) to the servers in order to access the virtual environment.

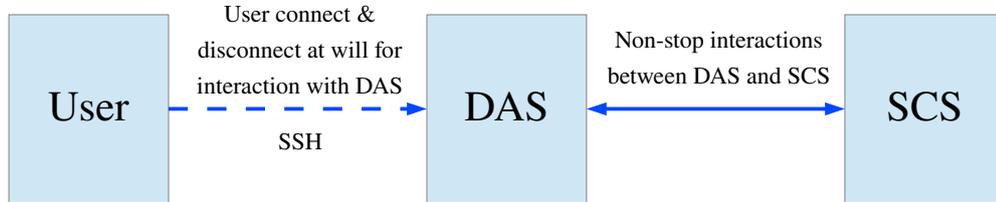


Figure 4.41: Block diagram showing the flow of interaction between the user, the Data Acquisition Software and the detection system.

#### 4.12.1 Slow Control interface

“Slow Control” is a term used in particular in particle physics and indicates an interface in charge of recording and logging all environment monitoring and subsystems interaction data for the user. Each set of measurements on the setup is taken every 30 seconds, with the interface being systematically refreshed at the same frequency. Figure 4.42 shows the *initialization* panel of the SC LabView interface. At the top of the window, the status of the overall instrumentation is reported in four boxes (colored in green if no errors, in flashing red otherwise), so that it is possible to check:

- the *illumination system* (monochromator and LEDs) together with photodiodes (for flux measurement) status: in this way it is possible to immediately check whether the related devices are turned on or not;
- the *acquisition* status, since when an acquisition starts, the DAS sends an acknowledgement signal to the SC;
- the *alarm* status;
- the *cryostat* status, since it is necessary to wait for the cryostat to be stable before the initialization of the acquisition.

Within the initialization panel there are several tabs and buttons which inform the user on the location where the data will be saved and logged (1), which instruments will be connected during the acquisition (2) and which starting parameter will be used for the next acquisition (3). If the user wants to stop the SC for any kind of reason, he has to press the red button indicated in Figure 4.42 with number 4. The second panel is the *vacuum* one and it mainly reports the pressure level within the cryostats (Figure 4.43). There are two pressure gauge mounted on the whole apparatus, one is on the cryostats and one is just at the entrance of the vacuum pump. The third panel is for the monitoring of the *thermal components* (Figure 4.44). In the section labelled with 1 there is the possibility for the user to select the operating temperatures

for the SCA and the SCE, then in tab 2 there is a report of the temperatures measured by the Lakeshore, in order to guarantee an accurate regulation of the two temperatures. In section 2 also the rate of temperature changing and the percentage of power that is delivered by the heaters are shown. In section 3 a monitoring over *TTi power supplies* is provided: as we have shown before, the cryostats are equipped with internal thermal copper shields in order to protect them from radiation; it is necessary to regulate the temperature of these thermal shields and hence the use of a TTI power supply system. In section 4 there are the temperatures coming from the components of what we call the Ni-DAQ, which is a set of instruments principally comprising Pt100s<sup>6</sup>. On each of the cryostats there are 12 Pt100s, Figure 4.45 shows where they are located. The fourth panel represents a display of the *Pt100 temperature measurements* as a function of time. The time axis is predisposed to show one point (corresponding to a temperature measurement) every 30 seconds (see Figure 4.46). The same type of display is provided for the optical components of the *illumination system* with the fifth panel (Figure 4.47). For the optical components both the display and the command options are provided, so that the user can command the illumination level provided by the LEDs and at the same time verify the measure registered by the 2 photodiodes that are placed in the cryostat (the precise positions of the LEDs and photodiodes are shown in Figure 4.45).

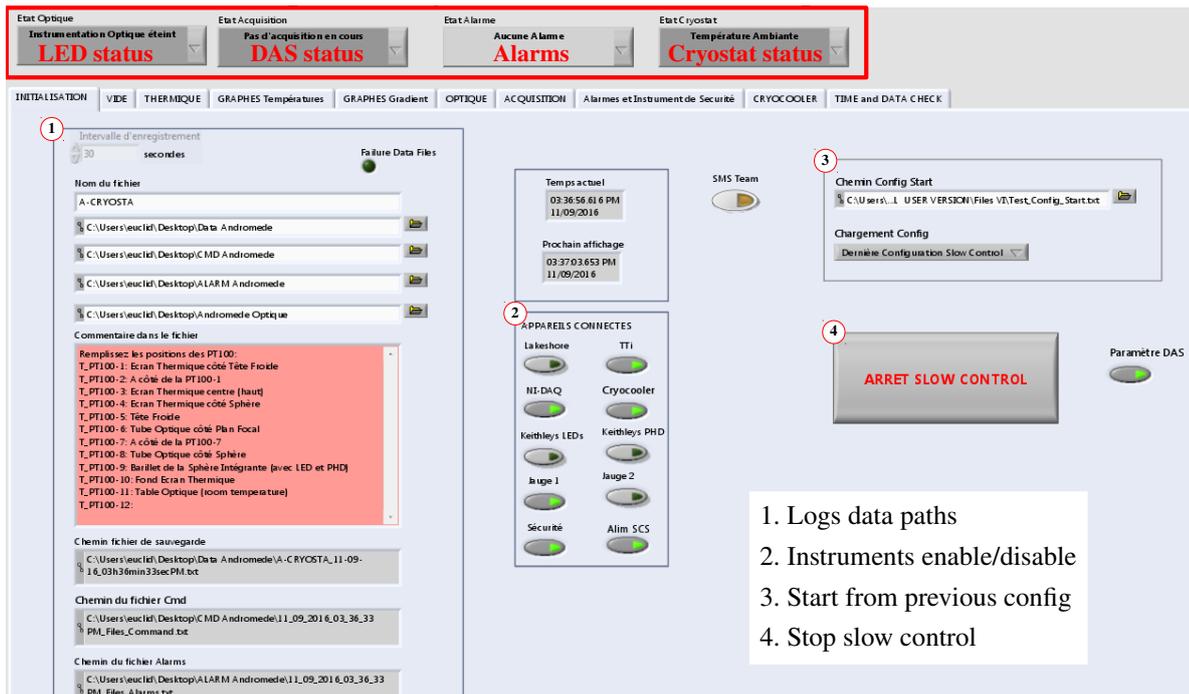


Figure 4.42: The initialization panel of the SC LabView interface.

<sup>6</sup>Pt100 are *resistance temperature detectors* (RTDs), i.e. sensors used to measure temperature. By far they are the most common RTDs used in industry and have a nominal resistance of 100 Ohms at 0 °C (“Pt” is the symbol for platinum, “100” stands for the resistance in Ohms at 0 °C).

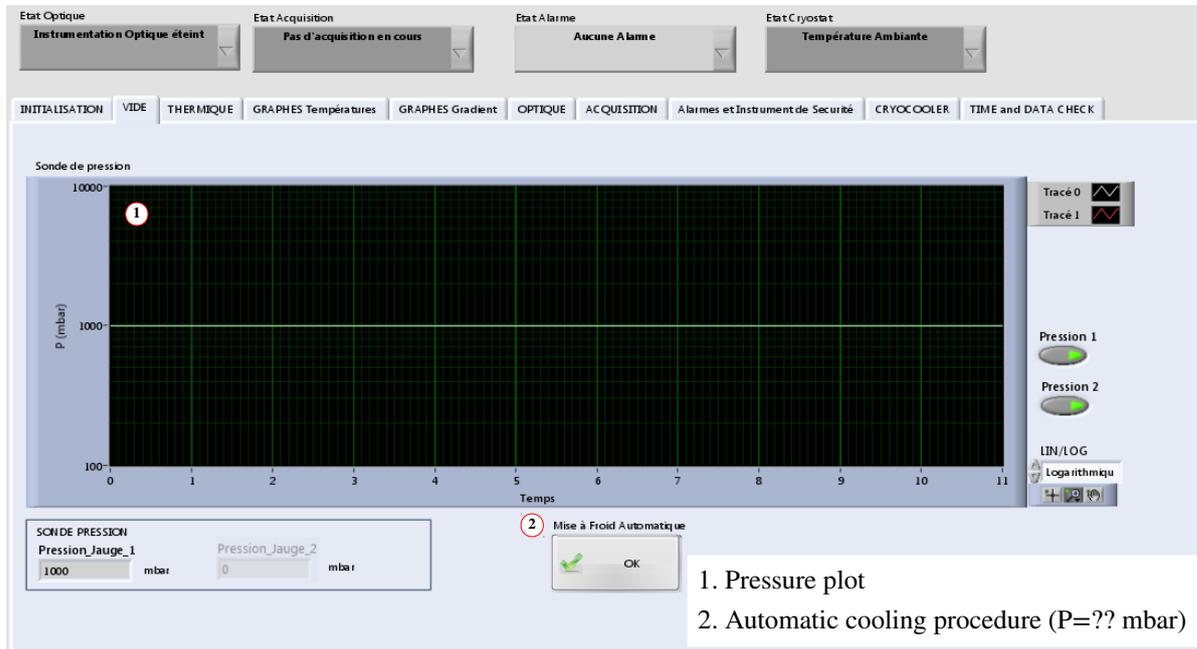
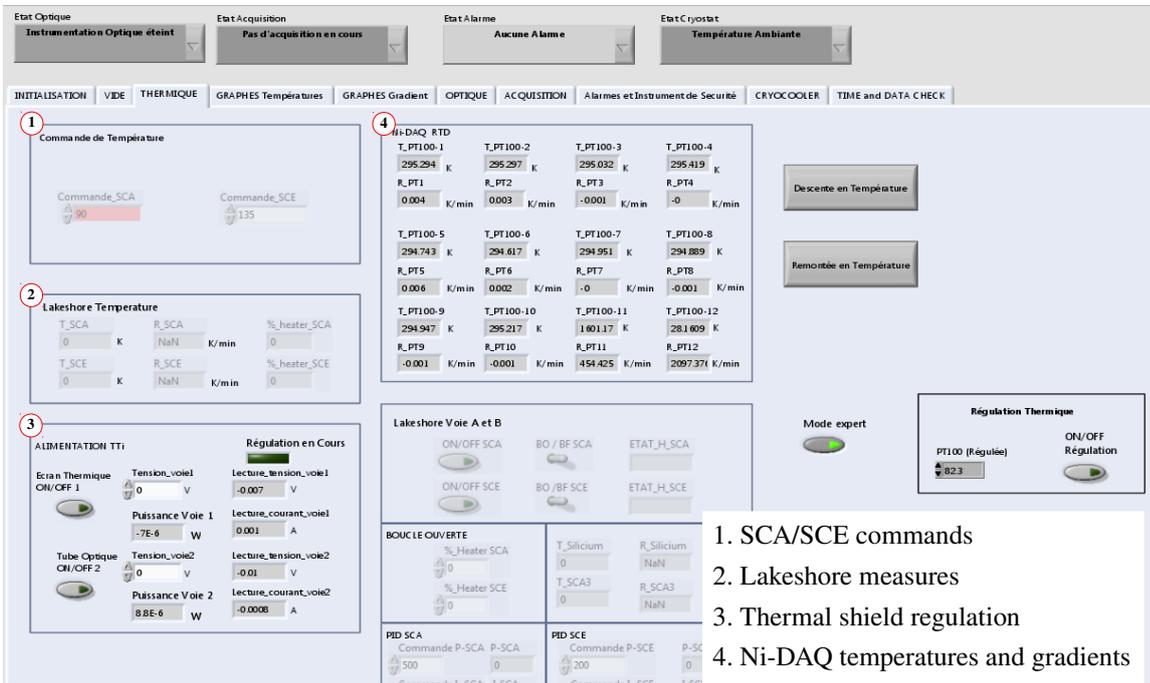


Figure 4.43: The vacuum panel of the SC LabView interface.

Commanding the LEDs illumination level basically means to apply a given current to the LEDs, then it is necessary to set the polarization bias for the photodiodes in order to measure the incoming light flux (1). If the measurement has to be taken with monochromatic light, the light flux is fixed: the user can just select the desired wavelength (3), then the monochromator selects automatically the grazing incidence of the mirrors to reach it and then opens the shutter to let the light coming out. In section 2 there are the displays with the evolution of the current for the photodiodes and with the evolution of current and voltage for the LEDs. The sixth and last relevant panel is the *acquisition* one (Figure 4.48). In this panel there is a section on the left named “Alim SCS”, corresponding to the commands useful to set the alimentation level of the detectors. On the right, the log of the communication between the acquisition software and the SC is provided: as a matter of fact, during an acquisition the DAS and the SC are exchanging information through JSON dictionaries, and the log allows to monitor the data which are being transmitted.

#### 4.12.2 The HDF5 acquisition file format

The output file format for an acquisition of the detectors is the HDF5 format [65]. As we have already mentioned, we define a sequence, a set of parameters which include environment and detector information (whether the LEDs are on or not, how many ramps have to be acquired, etc.). For the characterization it is necessary to collect a lot of statistics, so it is quite straightforward to define a variable which is called *cycle*, consisting of a given number of sequences that are repeated a given number of times (Figure 4.49). If, for example, we need to



1. SCA/SCE commands
2. Lakeshore measures
3. Thermal shield regulation
4. Ni-DAQ temperatures and gradients

Figure 4.44: The thermal components' panel of the SC LabView interface.

Pégase v7b  
30/03/2017

Andromède v7b  
30/03/2017

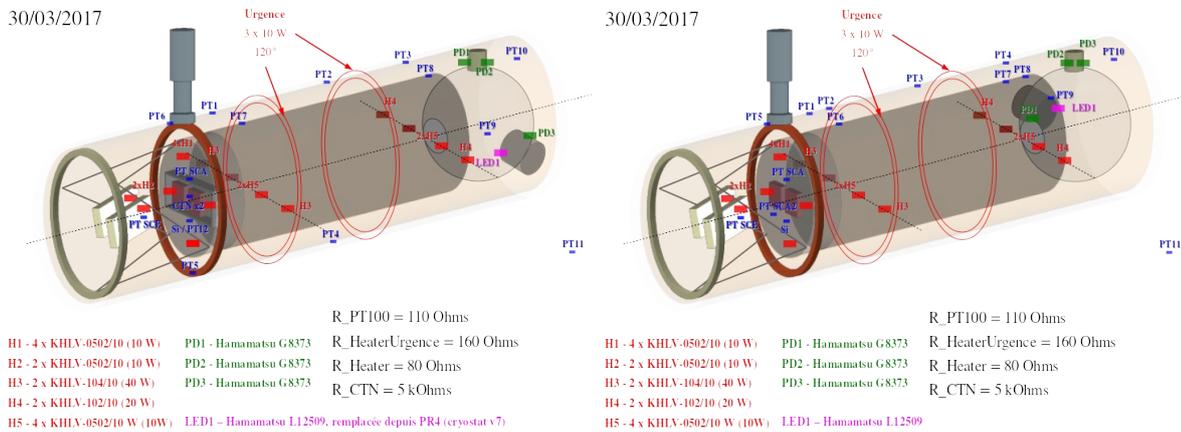


Figure 4.45: Left: Pt100 map of Pégase cryostat. Right: Pt100 map of Andromède cryostat.

do some sort of complicated measurement, i.e. something involving an arbitrarily alternating series of illumination and dark current measurements, we have to define two different sequences, one sequence with LEDs on, the other with LEDs off, and then we have to repeat them 10 times each. Instead of defining 20 sequences that have to be executed one by one, we just define

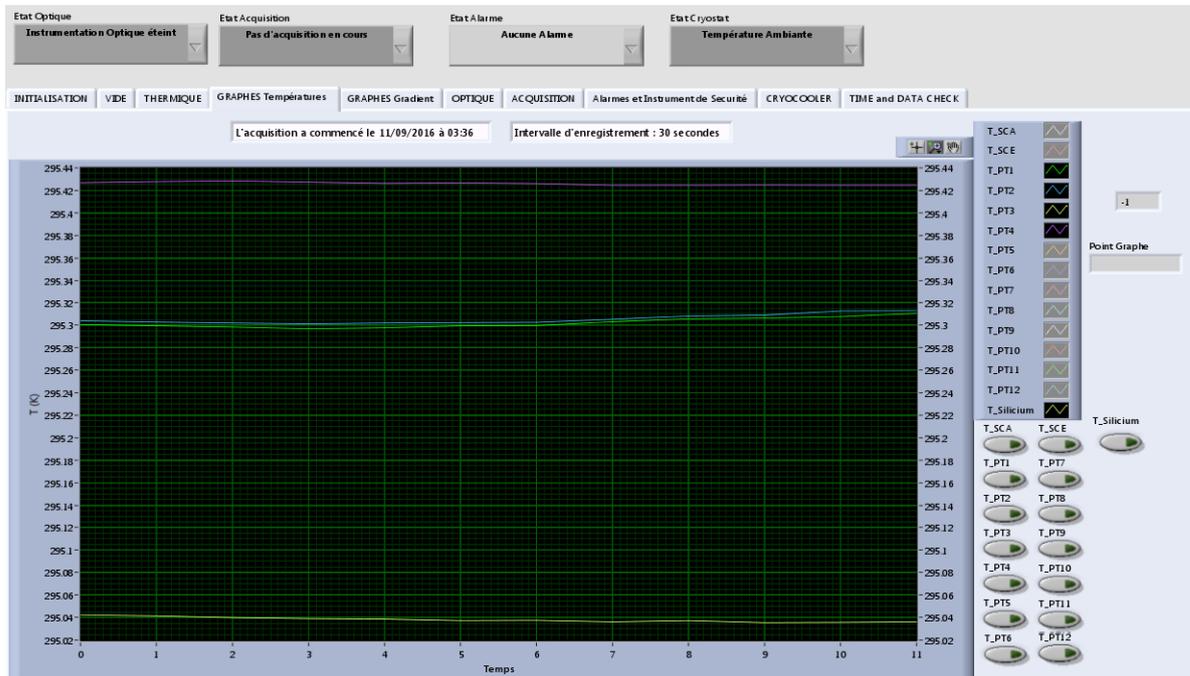


Figure 4.46: The Pt100 display panel of the SC LabView interface.

2 sequences, put them into a cycle and then repeat this cycle 10 times. The software used for displaying the architecture of a HDF5 file and the data stored in it, is called HDFView. When executed to open a HDF5 file, it shows the kind of structure reported in Figure 4.50. At first level we have what is called the *context* (1), which is a file to be defined prior to the acquisition in order to set in which context the acquisition will be made: as a matter of fact it includes the name of the cryostat, the id of the microcode which is loaded in the ASIC, the id of the LEDs, the version of the software and the hardware used. Then we have the DAS *dataset* (2), which is a dictionary containing all the configuration parameters for all the sequences. Finally we have a folder named *run* (3), which contains all the data recorded during the acquisition. In the run folder it is possible to see the different sequences defined. It has to be noticed that two of the sequences are called *init* and *end*: before and after the acquisition a brief check on the apparatus is done, so these are two sequences in which no data acquisition activity is carried out. A dataset called *env\_model* is present between the init/end sequences, which is in fact a dictionary listing all the parameters that are monitored by the SC, including also the file format in which the values of these parameters are stored. Figure 4.50 then shows what is contained in a *sequence* folder (4). Within a sequence there are two folders at the same level named *asic\_1* and *asic\_2*: since there are 2 ASICs per cryostat, for one sequence the data coming from both ASICs are acquired. As Figure 4.51 shows, in a *asic\_x* folder there is a given number of *ramp* folders (5), which include the acquired data. In a ramp folder there are the *frames* (the acquired images, indicated with number 6) and the *headers* (i.e. telemetry data, recorded values of the voltages applied on the detectors, etc., specified with number 7

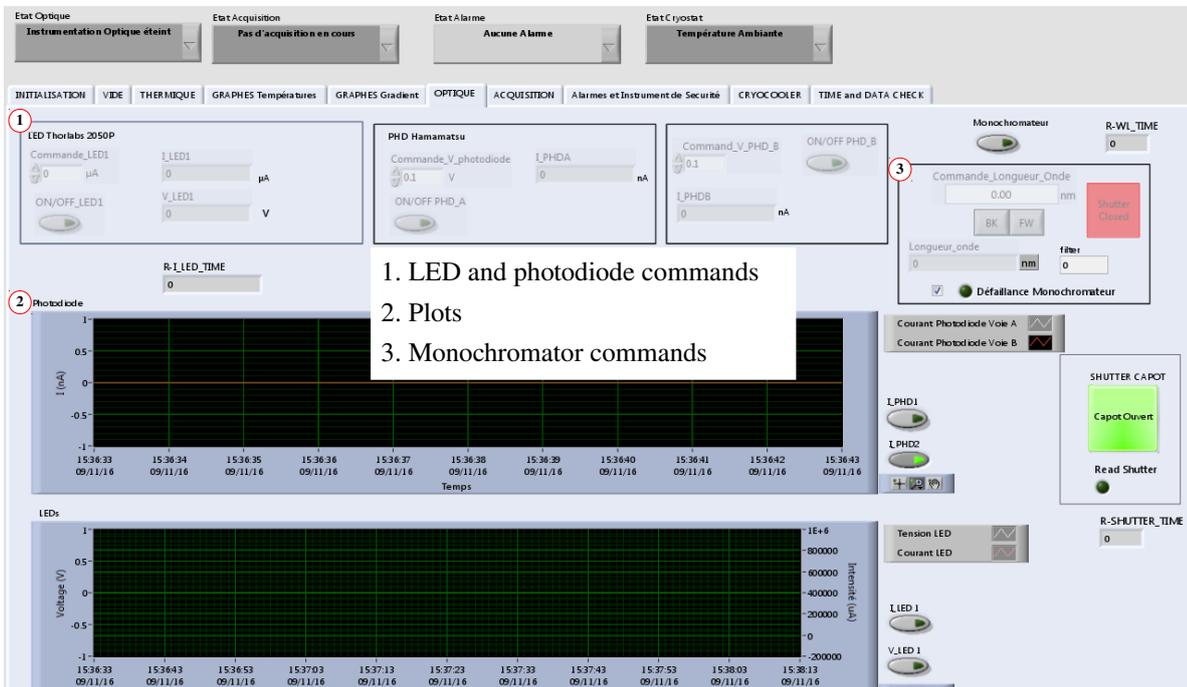


Figure 4.47: The illumination system panel of the SC LabView interface.

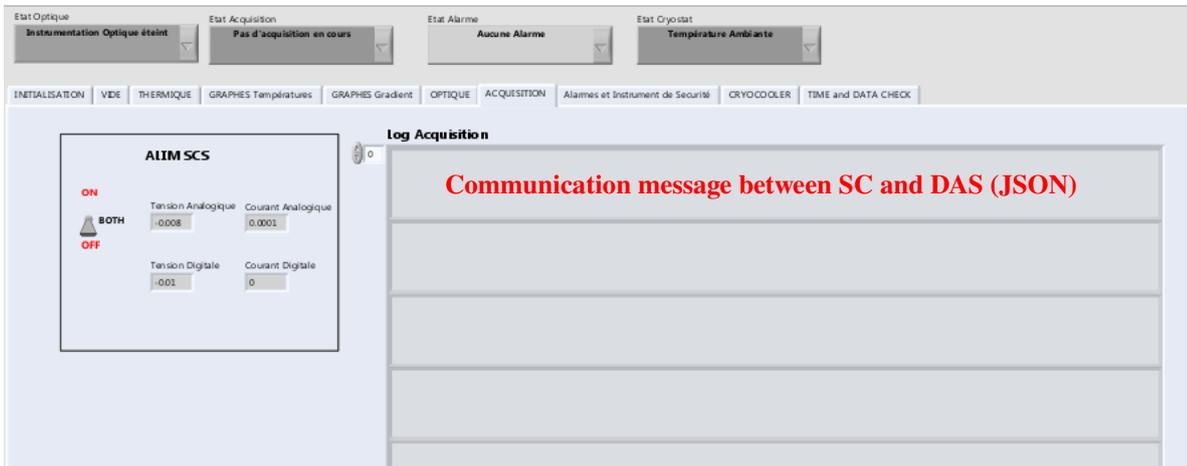


Figure 4.48: The DAS/SC communication log panel of the SC LabView interface.

in Figure 4.52). For each ASIC there are also the *sce\_config* (8) and *sce\_valid* datasets. The *sce\_config* shows all the microcode that has been loaded in the SCE, whereas the *sce\_valid* just contains a comparison before and after the loading of the microcode in the SCE (in order to check if the code is consistently loaded at the proper memory addresses). Concerning the

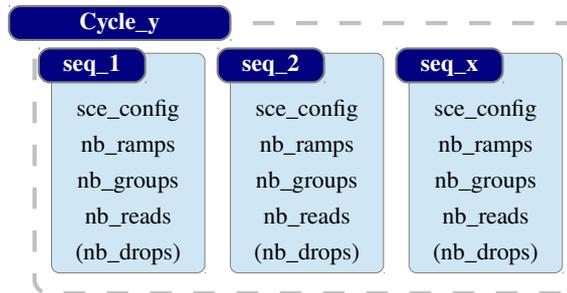


Figure 4.49: Illustration of a cycle containing a given number of sequences.

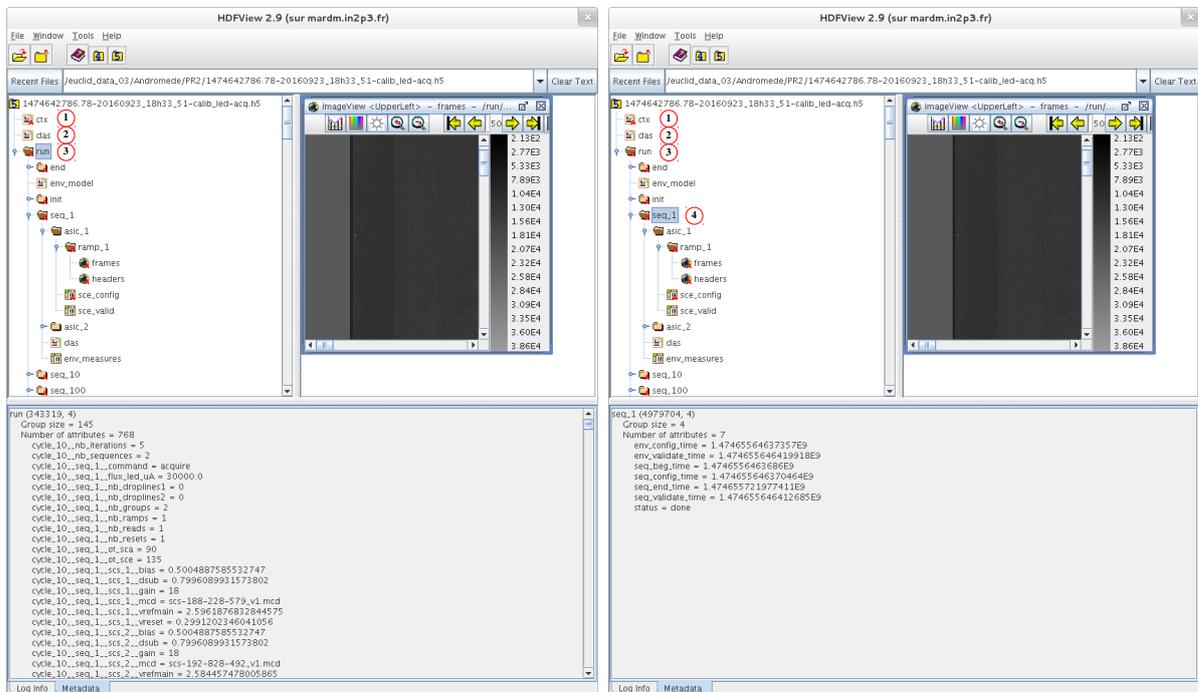


Figure 4.50: The structure of a HDF5 acquisition file displayed by HDFView. Left: the content of the run folder is shown. Right: the content of a sequence in the run folder is shown.

frames dataset (whose content is shown in Figure 4.51, on the right), a single H2RG detector has got a matrix of  $2048 \times 2048$  pixels, with another channel designed to acquire telemetry data (voltages and currents, temperature) during each frame. This channel is, as a standard SCE channel, of  $2048 \times 64$ , so that the total amount of data acquired for each image refers indeed to a  $2048 \times 2112$  pixel matrix.

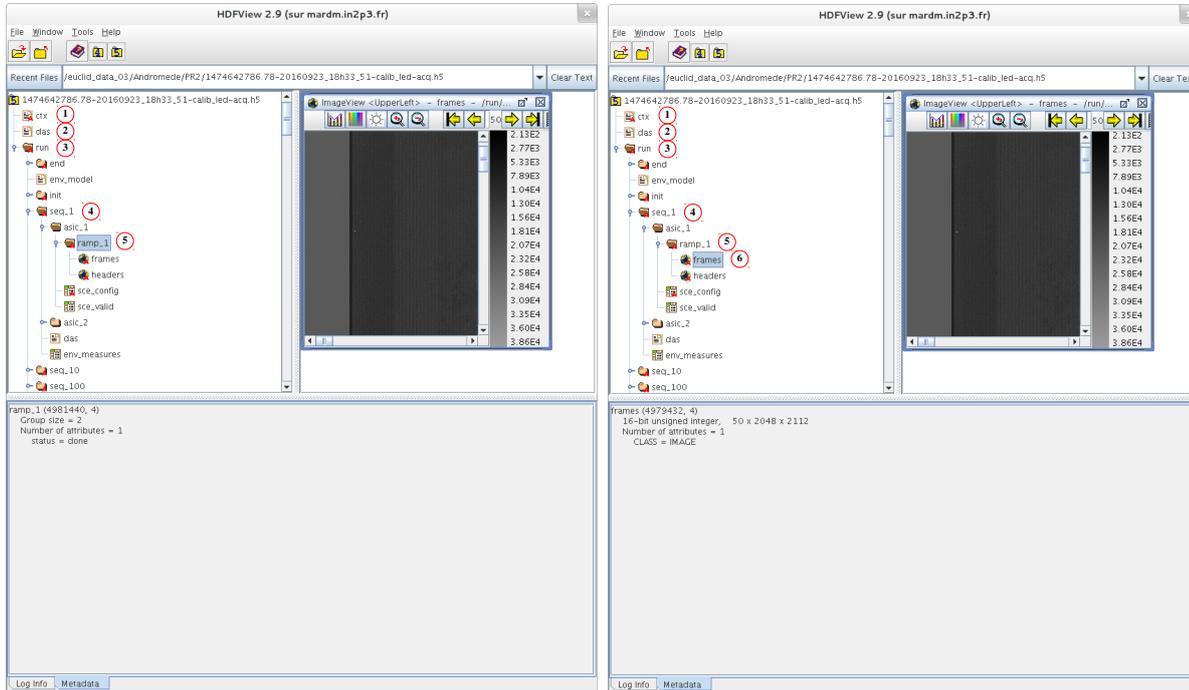


Figure 4.51: The structure of a HDF5 acquisition file displayed by HDFView. Left: the content of a ramp folder is shown. Right: the content of a frames dataset in the ramp folder is shown.

### 4.12.3 User interaction with DAS

As we have already seen, the DAS consists of a dedicated virtual environment on a Debian Unix server. The user can access the virtual environment through a tool called *tmux*. This software allows to create several sessions within a single terminal window, or rather to open several terminals within the same window, so that the *tmux* windows can be accessed from many devices at a time, providing the possibility, for example, of being helped by an expert looking at the same display. Once connected to the server via *tmux*, the user will find several open windows labelled with integer numbers (or with an optional name) and will be able to switch between them pressing the `ctrl + B +  $N_{window}$`  keyboard combination (with  $N_{window}$  the number of the selected window). As standard user, a shifter can access *mareuclid* server via *ssh* and then attach to a *tmux* session already created with the command `tmux attach -t [cryoname]` ([*cryoname*] is the name of the desired cryostat). At this point, the shifter can control two terminals: one terminal is opened on the computer just nearby the cryostat and provides access to the SC and to the managing of Markury electronics, while the other terminal is opened on *mareuclid* server. Usually the shifter has to deal with this second terminal, which allows the commanding and monitoring of the *workflow*. As a matter of fact, the *workflow* window opened in the terminal is the window that is used for the monitoring of the acquisition context: for example, it allows to check if the workflow is going on. Then there is the *monitor* window, which is used to monitor the communication status between the

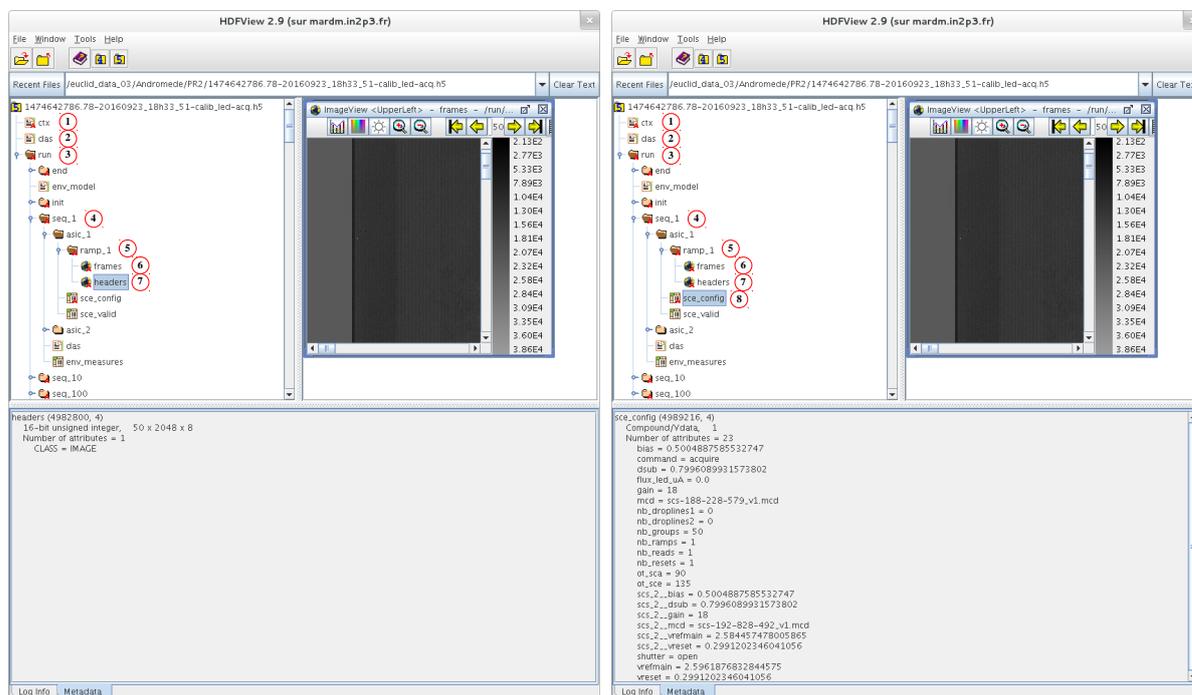


Figure 4.52: The structure of a HDF5 acquisition file displayed by HDFView. Left: the content of a headers dataset is shown. Right: the content of a sce\_config dataset is shown.

DAS and all the other systems (the SC, the Markury electronics, etc.): once a data exchange is done, this is printed on the monitor. Finally, there is the *command* window (*cmd*), which is used to launch the acquisition. Sometimes another window is opened just to launch the *data quality check* (which is described in §4.12.4). The virtual environment is specific to each of the two cryostats, so that, once connected to the server, the user needs to explicitly select the environment. To do this, for example, it is sufficient to type the command `workon fr1` (*fr1* stands for *flight run 1*) and then complete access to all the commands to control the cryostats is provided. On the *workflow* window it is possible to see the data that are going to be acquired: it is sufficient to type the command `[cryoname]-cmd workflow.list` in order to see the list of data. With the command `[cryoname]-cmd workflow.status` the user can check what is going on during the acquisition: if no acquisition is running, the command will print a small dictionary on the screen with the specification that the electronics is in *idle* mode, while if there is an acquisition in progress, a message telling which acquisition is being executed and which sequence is running will be shown. When the user has no idea of the process that is being executed, he has just to type `[cryoname]-cmd workflow.status` in order to find out. The command to execute a particular acquisition is `[cryoname]-cmd workflow.execute [acqname]` (where `[acqname]` is the acquisition procedure name), whereas `[cryoname]-cmd workflow.abort` and `[cryoname]-cmd we.power_off` respectively stop the acquisition progress and the electronics directly (see Figure 4.53). We will provide here a basic

		acqnames
<b>In tmux command window only, replace [cryoname] by pegase or andromede</b>		
<b>Initialize warm electronics &amp; detectors</b>	[cryoname]-cmd we.power_on	interconnect
<b>View actual workflow</b>	[cryoname]-cmd workflow.list	initial_check
<b>View status</b>	[cryoname]-cmd workflow.status	baseline
<b>Run all workflow</b>	[cryoname]-cmd workflow.execute all	dark_long
<b>Run specific acq.</b>	[cryoname]-cmd workflow.execute [acqname]	dark_science
<b>Abort acquisition</b>	[cryoname]-cmd workflow.abort	latency_low_flux
<b>Shutting down electronics</b>	[cryoname]-cmd we.power_off	nl_low_flux
		nl_med_flux
		nl_hig_flux
		latency_hig_flux

Figure 4.53: The list of commands to type in order to command and monitor the workflow on the DAS virtual environment. On the right, a list of the acquisition procedures.

description of the acquisition procedures listed in Figure 4.53:

- *interconnect*: test at room temperature in order to see the number of pixels that are connected and measuring;
- *initial check*: acquisition made when the detector is cold, in order to check whether the baseline is correct;
- *baseline*: test to monitor the amount of the offset at the beginning of the ramp;
- *science dark*: acquisition to measure the dark current, organized in 50 ramps of 500 frames;
- *long dark*: acquisition to measure the dark current, organized in 4 ramps of 8000 frames;
- *non-linearity*: acquisition to measure the non-linearity effects (with indication of the flux in  $e^-/s$ );
- *latency*: acquisition to measure the latency effects (with indication of the flux in  $e^-/s$ ).

## Logbooks

When an acquisition is completed, all the related information (about the occurred events and the results) is reported in logbooks. The logbook is filled by a person on-call, whom the shifter has to contact if something goes wrong during the acquisition progress. All the logbooks are hosted on a OneDrive repository with limited access: there are specific logbooks for the flight runs and one logbook per cryostat is maintained. There is also the possibility to use a Python script to automatically generate logbooks starting from an ASIC dataset, and usually when a flight run is over, the corresponding logbooks are automatically generated in this way in order to immediately check data consistency.

#### 4.12.4 Data quality check (QC)

The acquisition data are organized in 2 levels:

1. *L0 data*, which are raw data managed by the DAS and contained in the HDF5 file previously seen;
2. *L1 data*, which are data produced by a quality check program running with L0 data in input: this quality check is a kind of preliminary data analysis aiming to validate the goodness of the data taking;

As a shifter, the user has just to control if quality check parameters contained in the analysis report are within thresholds, and for this purpose there is a web portal to access with a username and a password where to consult the HTML page with the L1 data. The SC actually records environmental data and then transmits them to the same HTML page every 30 seconds. L1 data on the HTML page are refreshed every 3 hours.

#### 4.12.5 Analysis report

The analysis report mainly contains a list of parameters which are relevant for the characterization of the H2RG detectors. Each parameter presents its own value derived from the acquisition data, and each value must lay within a specific range in order to guarantee the optimal detector performance. As Figure 4.54 shows, the result of the acquisition is a classification of the parameter values according to 3 tags: *Passed*, *Warning* or *Failed*. It is quite easy to understand that when the number of *Failed* parameters becomes large, some kind of action has to be taken.

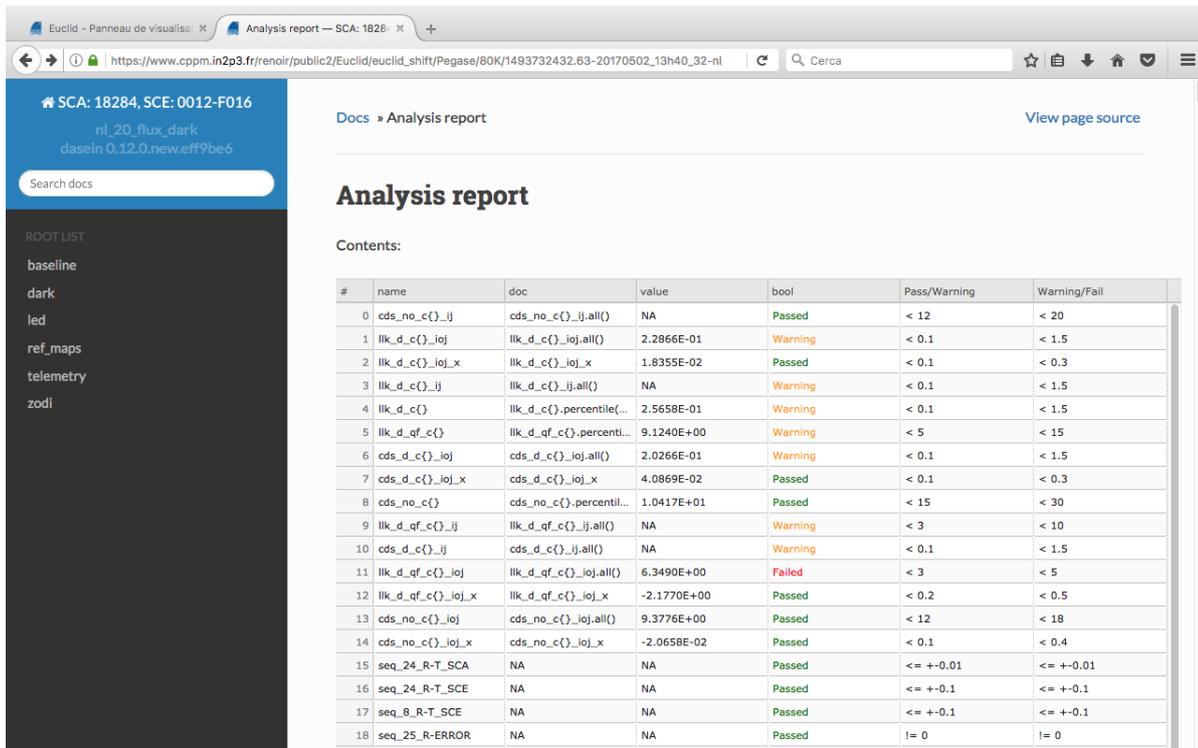


Figure 4.54: The HTML page showing the analysis report. On the left, a ROOT LIST is visible, which allows to select the category of acquisition data to be analyzed in detail.

From the ROOT LIST on the left it is possible to select one of the specific acquisition procedures and to analyze the related data in detail. For example, Figure 4.55 illustrates the available characterization data for a *baseline* acquisition. A brief legend is reported in Table 4.6 in order to understand the conventions used for data naming. Each data link shown in the *baseline list* (and, in general, in a sub-list of the ROOT LIST) mainly provides 3 different analytical perspectives over the corresponding data distribution: *Histogram*, *Colormap* and *Statistic* (see Figure 4.56 representing the *kTC noise* data example). The task of a shifter is to check that relevant data values for a specific acquisition procedure are compliant with the constraints given by Table 4.7. Moreover, the shifter has also to monitor and report the values of the pressure in the cryostat, the temperature of the SCA and the temperature of the LEDs (the information being available on the custom web portal or on the SC LabView interface).

<sup>7</sup>The sky background in the field of view is called the *zodiacal background*. It will be dispersed due to observation in slitless spectroscopy, and all the associated wavelengths are integrated on the pixels.

The screenshot shows a web application interface for monitoring characterization data. The browser address bar indicates the URL: [https://www.cppm.in2p3.fr/renoir/public2/Euclid/euclid\\_shift/Pegase/80K/1493732432.63-20170502\\_13h40\\_3/](https://www.cppm.in2p3.fr/renoir/public2/Euclid/euclid_shift/Pegase/80K/1493732432.63-20170502_13h40_3/). The page title is "baseline — SCA: 18284, SCE: 0012-F016".

The main content area is titled "baseline" and contains a table with the following data:

#	name	doc	value	bool	Pass/Warning	Warning/Fail
0	b	b.percentile(2)	1.0335E+04	Passed	> 5000	> 3000
1	b_ij	b_ij	NA	Passed	> 5000	> 3000
2	b_ij_x	b_ij_x.min	-1.8947E-01	Failed	> -0.05	> -0.10
3	ktc_loj	ktc_loj	4.7635E+01	Warning	< 40	< 50
4	ktc_loj_x	ktc_loj_x	-6.5477E-02	Passed	> -0.10	> -0.25
5	b_loj	b_loj	1.5383E+04	Passed	> 5000	> 3000
6	b_loj_x	b_loj_x	-1.7127E-01	Failed	> -0.05	> -0.10
7	ktc_ij	ktc_ij	NA	Failed	< 40	< 50
8	ktc_ij_x	ktc_ij_x.min	-1.4023E-01	Warning	> -0.10	> -0.25
9	b_high	b_high	2.7080E+04	Failed	> 65000	> 60000
10	b_high_x	b_high_x.abs()	2.5394E+04	Failed	< 500	< 2000
11	b_low_x	b_low_x.abs()	2.1541E+03	Failed	< 500	< 2000

Below the table is a "baseline list" section with a bulleted list of items:

- b
- b\_chrono
- b\_high
- b\_ij
- b\_loj
- b\_low
- ktc
- ktc\_ij
- ktc\_loj

The interface also features a sidebar on the left with a "ROOT LIST" and a search bar. The sidebar contains a list of items: b, b\_chrono, b\_high, b\_ij, b\_loj, b\_low, ktc, ktc\_ij, ktc\_loj, dark, led, ref\_maps, telemetry, and zodi.

Figure 4.55: The characterization data available from the ROOT LIST in the case of a baseline acquisition.

<b>Short name</b>	<b>Complete name</b>	<b>Meaning</b>
<b>b</b>	<i>baseline</i>	first frame after reset
<b>cds</b>	<i>correlated double sampling</i>	ramp slope calculated by subtraction of 2 frames
<b>llk</b>	<i>likelihood</i>	ramp slope calculated with likelihood method
<b>lsf</b>	<i>least square fit</i>	ramp slope calculated with least square fit
<b>ktc</b>	<i>reset noise</i>	baseline standard deviation
<b>d</b>	<i>dark current</i>	ramp slope with LEDs off
<b>no</b>	<i>noise</i>	ramp noise with LEDs off
<b>qf</b>	<i>quality factor</i>	likelihood quality factor ( $\chi^2$ )
<b>x</b>	<i>cross-check</i>	comparison with reference pixel maps
<b>l</b>	<i>illumination</i>	gross illumination
<b>led</b>	<i>corrected illumination</i>	illumination with dark current subtraction
<b>flat</b>	<i>corrected illumination</i>	illumination with dark current subtraction and quantum efficiency correction
<b>zodi</b>	<i>illumination</i>	illumination simulating the zodiacal background <sup>7</sup>
<b>i</b>	<i>rows</i>	average on columns
<b>ij</b>	<i>channels</i>	average on rows and columns by channel
<b>oj</b>	<i>columns</i>	average on rows
<b>ioj</b>	<i>matrix</i>	average on rows and columns
<b>g</b>	<i>groups</i>	average on the first 16 frames
<b>chrono</b>	<i>time evolution</i>	one point per ramp
<b>c1/c2</b>	<i>cycle (1 or 2)</i>	number of acquisition cycle

Table 4.6: The naming conventions used for quality check data in the analysis report.

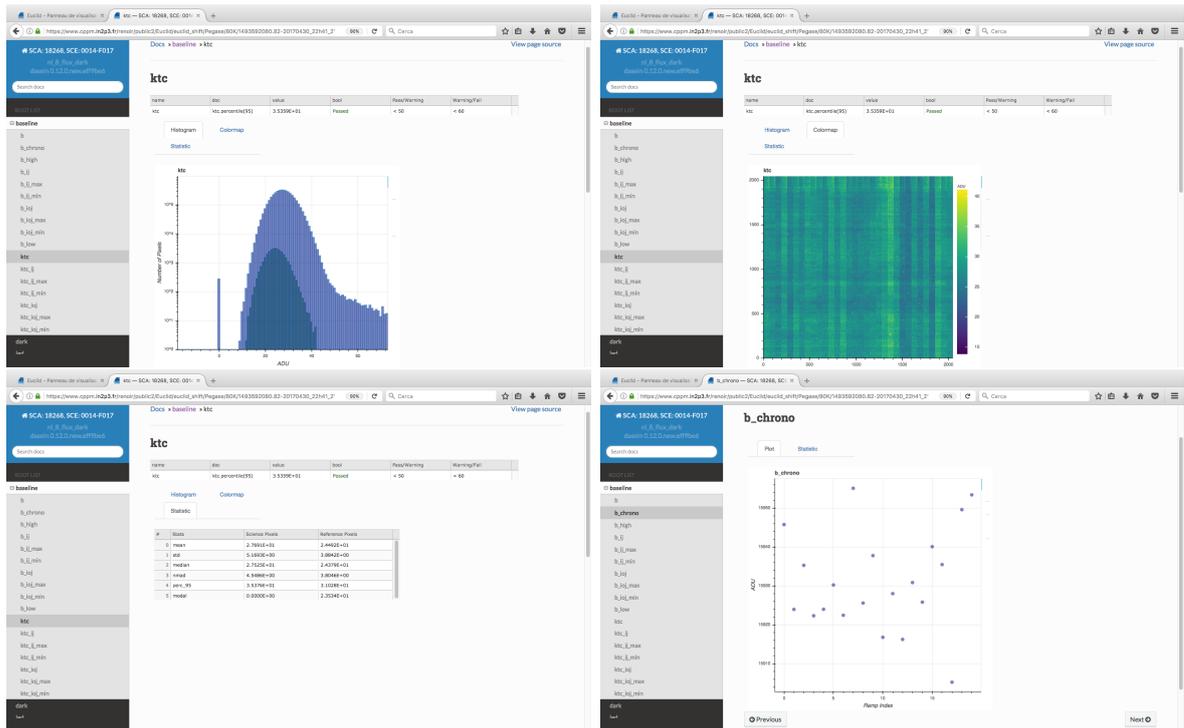


Figure 4.56: Illustration of the kTC noise data analytical perspectives: at top left, the histogram with the pixel values distribution (in purple science pixels, in green reference pixels); at top right, the colormap with the matrix of values pixel per pixel; at bottom left, a table summarizing relevant statistical estimators. At bottom right, the chronogram of the baseline is reported as an example. We recall that  $1 \text{ ADU} = 2 e^-$  (gain = 15 dB).

Acquisition	ROOT LIST	Content checks	Logbook report	Oddities	To report
<b>zodi</b>	zodi	The value of the zodi flux must always be the same, around 1-1.5 ADU/frame; the chronogram of the flux must be stable over all the ramps.			
<b>science &amp; long dark</b>	baseline	The chronogram of the baseline must be stable (max 50 ADU of variation); the map of the ktc must be proper.			
	dark	The value of the flux of the dark (cds or llk) must be of the order of $10^{-2}$ ADU/frame; the chronogram of the flux must be stable on all the ramps; the value of the cds noise must be of the order of 9-10 ADU; the chronogram of the noise must be stable over all the ramps.	Median of cds_d (dark flux value).	Bad pixels/hot pixels seen on ij/oj; a channel more noisy than the others.	Light leaks on the colormap.
<b>non-linearity &amp; latency</b>	baseline	The chronogram of the baseline must be stable (max 50 ADU of variation); the map of the ktc must be proper.		Vertical burrs on ktc map.	
	dark	The flux value of the dark (cds or llk) is related to persistence; the value of the flux of the dark must be consistent with the values observed previously; the chronogram of the dark tends to evolve because of the persistence.	Median of cds_d.c2 (dark flux value).	The colormap shows the effects of persistence.	
	led	The flux of the LED must be close to requested value (factor 1.4 between ADU/frame and $e^-/s$ ); the flux obtained by the two detectors must be similar; the chronogram of the LED should be stable; the map must be homogeneous ( $< 3\%$ ).	Median of llk_l.c2 (LED flux value).	Chronogram increase on n.l., decrease on latency; flat badly corrected by q.e./baseline visible on the map.	
	zodi	The zodi flux must be around 1-1.5 ADU/frame; the flux chronogram must be stable over all the ramps.	Median of llk_zodi.c1 (zodi flux value).	Chronogram increase on n.l.	

Table 4.7: Table summarizing the list of checks demanded to the shifter during the characterization.

---

H2RG dark current analysis

---

In this chapter I describe the analysis carried out to study the dependence on temperature of H2RG dark current during the detector characterization campaign (referred to as Flight Run 2, FR2). FR 2 took place from September 6th to October 25th 2017, involving 4 Sensor Chip Systems (SCS), two located in the *Pégase* cryostat and the other two placed in the *Andromède* cryostat (see Table 5.1). The aim of the characterization campaign is to test the response of the H2RG detectors at pixel level so as to calibrate and to verify the performance of the various NISP subsystems, according to the diagram shown in Figure 5.2.

Cryostat	Position	SCA id	CFC id	SCE id
<i>Pégase</i>	P1	18268	045	019
	P2	18284	081	017
<i>Andromède</i>	P1	18249	014	015
	P2	18278	015	043

Table 5.1: Table showing the components of the SCS used for FR 2 calibration campaign. Positions P1 and P2 refer to Figure 5.1.

## 5.1 Dataset description

The current analysis was done considering the data acquired from *Pégase* SCAs, 18268 and 18284. Correspondent SCEs are named ASIC 1 and ASIC 2, respectively. The detector dark current is measured according to two different methods:

- *science dark*, corresponding to the acquisition of a sequence of 50 signal ramps each made of 500 frames. This acquisition pattern is coherent with the maximum number of frames per exposure (500) that NISP will acquire in flight;

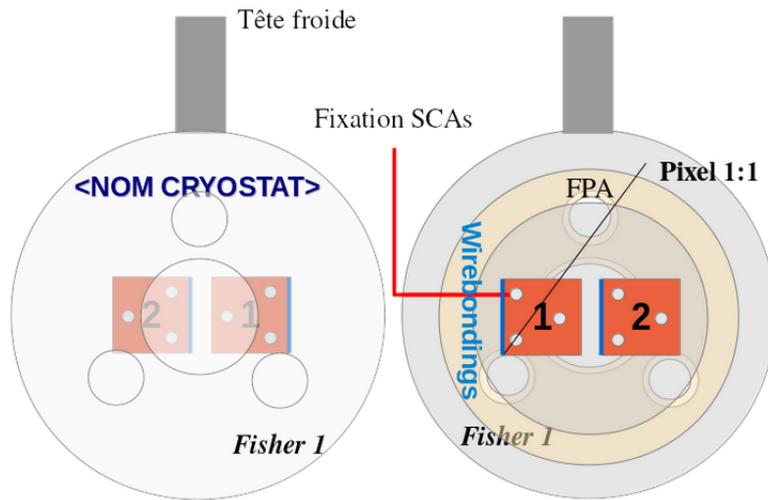


Figure 5.1: The Sensor Chip Assemblies disposition on the focal plane arrays of Pégase and Andromède.

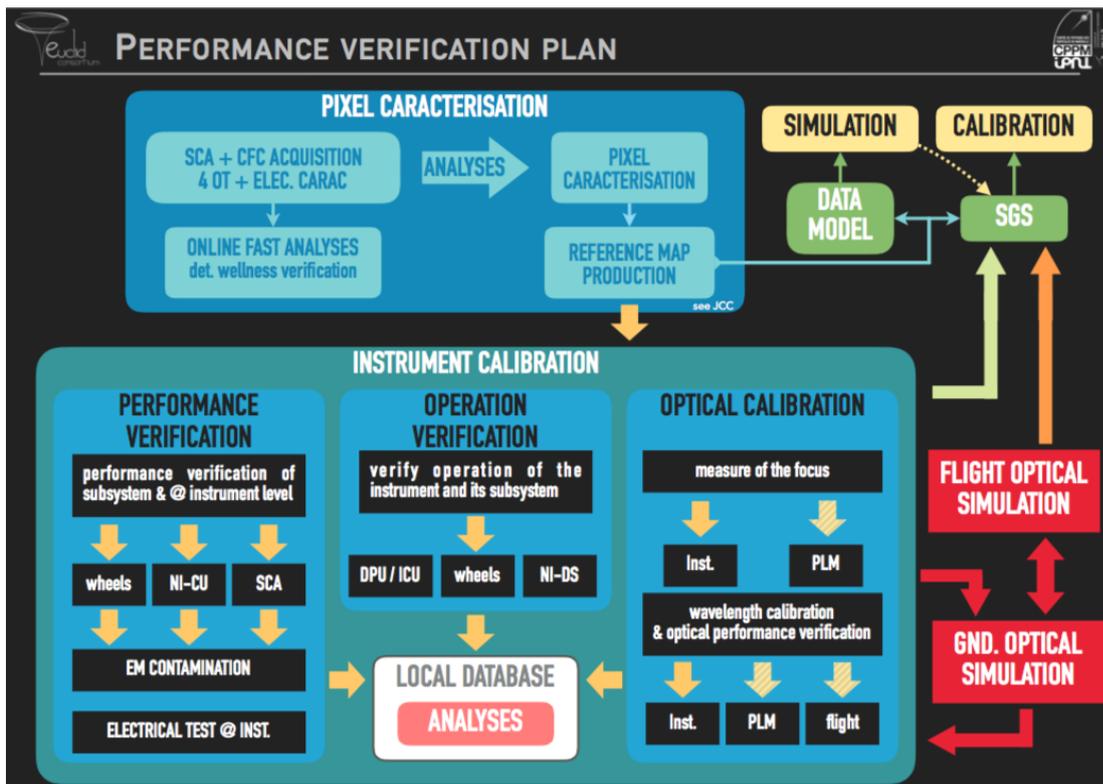


Figure 5.2: Diagram showing the phases of the NISP subsystems' performance verification plan.

- *long dark*, which refers to an acquisition consisting of 4 signal ramps each made of 8000 frames. This mode is useful to study the effects of persistence on dark current.

Since “science dark” is the mode which better reproduces NISP in-flight operation, it was the one chosen to implement the dark current analysis. The final goal of the analysis is to assess the thermal dependency of the SCAs for further in-flight correction. The dataset consisted of quality check L1 data (introduced in §4.12.4) taken from acquisitions done at 80, 85, 90 and 100 K. The data are stored in HDF5 files (see §4.12.2) that are converted into FITS images (see Appendix A) and then into CERN ROOT files. The dark current of H2RG detectors is generally dominated by diffusion effects due to thermally generated carriers from the bulk. Its dependence on temperature can be described by the *Arrhenius law*  $i_{diff} \propto e^{-\frac{E_{bg}}{kT}}$ , with  $E_{bg}$  the bandgap energy,  $k$  the Boltzmann constant and  $T$  the absolute temperature of the detector [66]. However, at low temperatures ( $T \leq 100$  K) the dominating effect is related to trap-assisted tunneling (Figure 5.3) and small leakage currents, so that a constant term  $i_{const}$  in the functional dependence of dark current on temperature must be added. For these reasons, the fit function considered in the analysis was

$$i_{dark}(T) = i_{\infty} e^{-\frac{E_{bg}}{kT}} + i_{const}. \quad (5.1)$$

$i_{\infty}$  approximately corresponds to the value of dark current at ideal infinite temperature; as a matter of fact  $i_{dark}(\infty) = i_{\infty} + i_{const}$ .

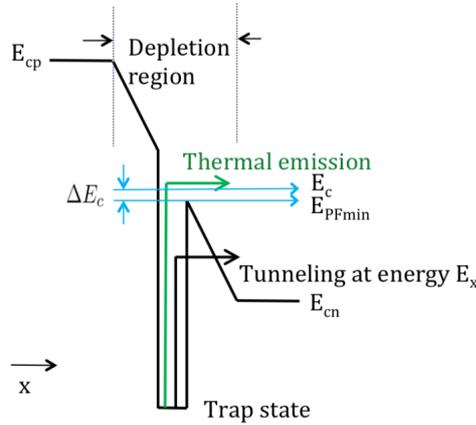


Figure 5.3: A schematic diagram of the trap-assisted tunneling: an electron can pass from the valence band to the conduction band through the combination of the absorption of a phonon and a tunneling process [67].  $E_c$  is the conduction band energy within the depletion region,  $E_{cp}$  and  $E_{cn}$  are the conduction band energy levels in the P and N doped sides, respectively, and  $E_{PFmin}$  accounts for the Poole-Frenkel effect, i.e. the lowering of the electron barrier due to the Coulomb interaction of the trap with the lattice.

## 5.2 Pixel selections

Dark current data for each detector pixel were obtained applying the  $\chi^2_{Euclid}$  flux estimator to the acquired signal ramps (see §4.7.4). The dark current for a given H2RG detector at a specific absolute temperature was computed by averaging the dark current values over the whole  $2048 \times 2048$  pixel matrix. The associated error is the standard error on the mean of the distribution,  $\sigma_\mu = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N(N-1)}}$ , where  $x_i$  is the dark current of the  $i$ -th pixel,  $\mu$  is the average value of dark current over the whole matrix and  $N = 4194304$  is the total number of pixels.

Figure 5.4 - top panel, shows the average dark current of the 32 readout channels of the detector (Figure 4.16) as a function of  $1/(kT)$  for ASIC 1 (SCA 18268 & SCE 019); Figure 5.4 - bottom panel, shows the same plot for ASIC 2 (SCA 18284 & SCE 017). These plots show the presence of outlier values; in order to exclude anomalous pixels, a filter was applied based on suitable statistical indicator. The average dark current for each readout channel was considered. The distributions of dark current over the whole matrix at different temperatures are shown in Figure 5.5 and Figure 5.6, for ASIC 1 and ASIC 2, respectively. They are far from gaussian, with high values of skewness and kurtosis (Table 5.2), having longer tails at higher dark current values and rather sharp central peaks. The t-Student's distribution generated by the variable  $t_i = \frac{\bar{X}_i - \mu}{S_i/\sqrt{n}}$  was considered.  $\bar{X}_i$  is the mean value of dark current for the  $i$ -th channel,  $\mu$  is the dark current mean value over the whole matrix,  $S_i$  is the standard deviation for the dark current distribution of the  $i$ -th channel and  $n$  is the number of samples, i.e.  $n = 32$ . We selected channels within specific distance from the mean values of t-Student's distributions in order to minimize the figure of merit  $FoM = |\chi^2/ndf - 1|$ .  $\chi^2/ndf$  is the normalized  $\chi^2$  of the fit to the Arrhenius law of dark current mean values over the whole matrix. t-Student's distributions are shown in Figure 5.8 and Figure 5.9.

At each temperature we excluded channels with dark current average value external to the interval  $[\mu_t - c \cdot \sigma_t, \mu_t + c \cdot \sigma_t]$ , with  $\mu_t$  the mean of the t-Student's distribution,  $\sigma_t$  its standard deviation and  $c$  an arbitrary factor. In Figure 5.7 the  $\chi^2/ndf$  of the fit to the Arrhenius law versus  $c$  for ASIC 1 and ASIC 2 is shown. The value of  $c$  which minimizes the  $FoM$  was chosen. For ASIC 1,  $\chi^2/ndf = 1.022$  when  $c = 0.8$ , as one can see in Figure 5.7 - top panel. Channels 1, 2, 3, 4, 5, 7, 9, 10, 11, 14, 17, 18, 19, 22, 23, 24, 29, 31, 32 being external to the interval  $[\mu_t - 0.8\sigma_t, \mu_t + 0.8\sigma_t]$  were excluded. After the selection only 1703936 pixels (40.6 %) survive. The corresponding Arrhenius plot is shown in Figure 5.10. For ASIC 2, Figure 5.7 - bottom panel, shows that, by excluding from the t-Student's distribution a larger number of channels, the  $\chi^2/ndf$  gets closer to 1 until  $c = 1.4$  and then it gradually decreases for  $c < 1.4$ . The  $FoM$  reaches its minimum when  $c = 1.4$ . This selection excludes channels number 1, 2, 3, 4, 5, 8, 32. After the selection 3276800 pixels (78.1 %) survive. The Arrhenius plot for ASIC 2 obtained after the selection is shown in Figure 5.11.

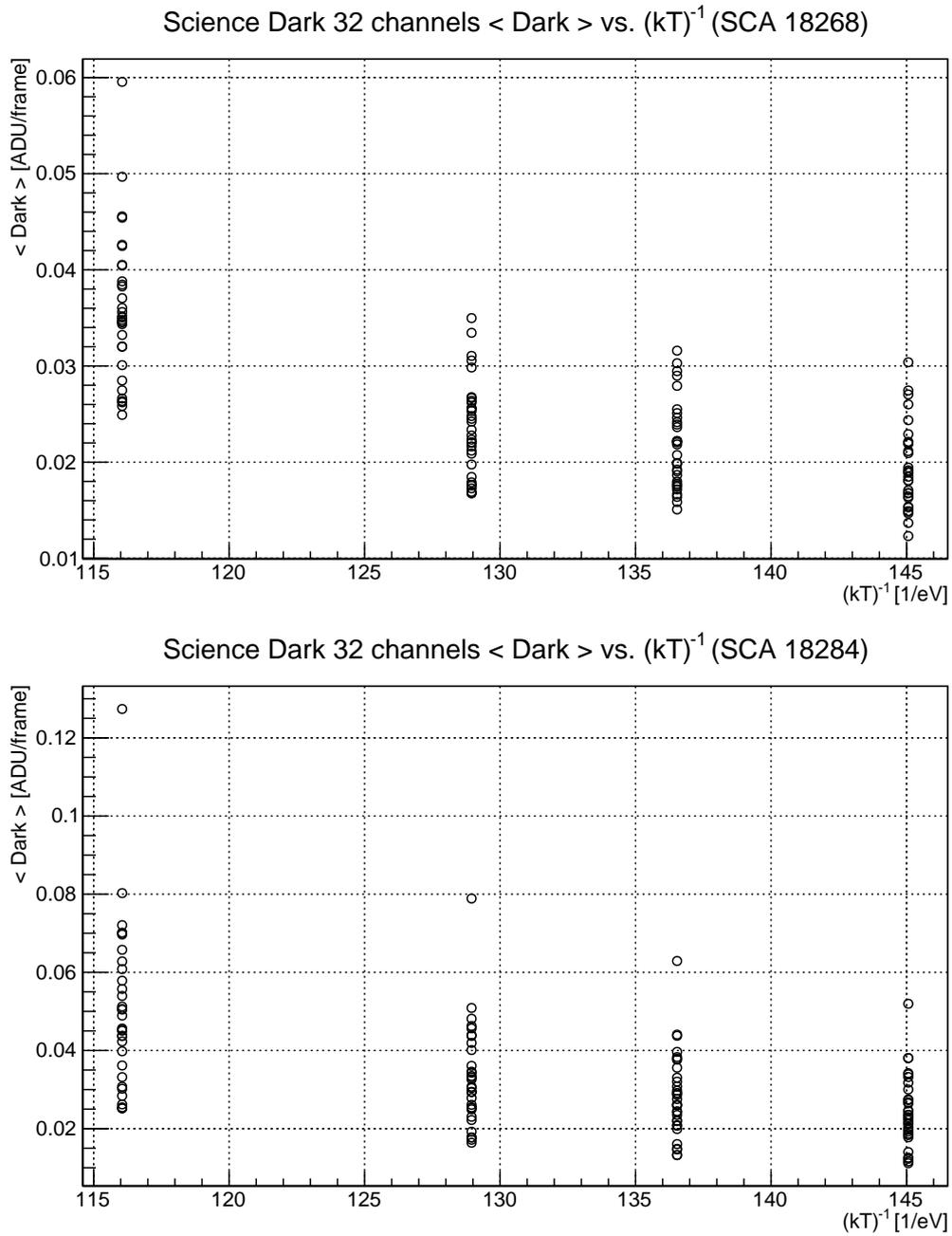


Figure 5.4: Plot showing the mean dark current of 32 detector channels as a function of temperature for ASIC 1 (top panel) and ASIC 2 (bottom panel).

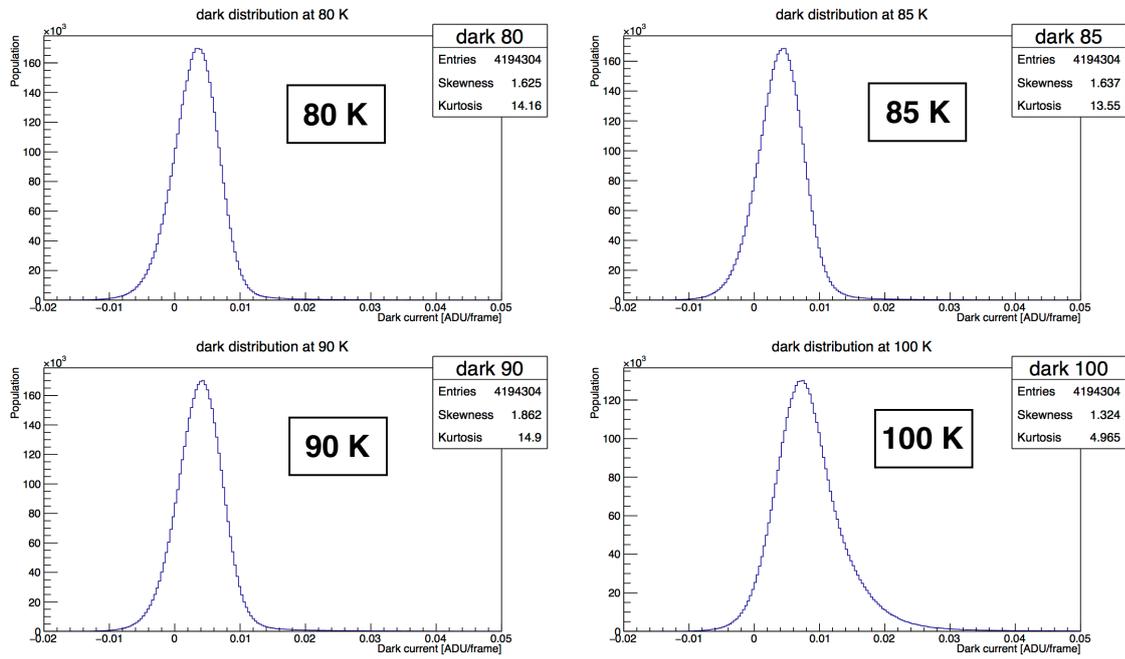


Figure 5.5: Dark current distributions for ASIC 1 at 80, 85, 90 and 100 K.

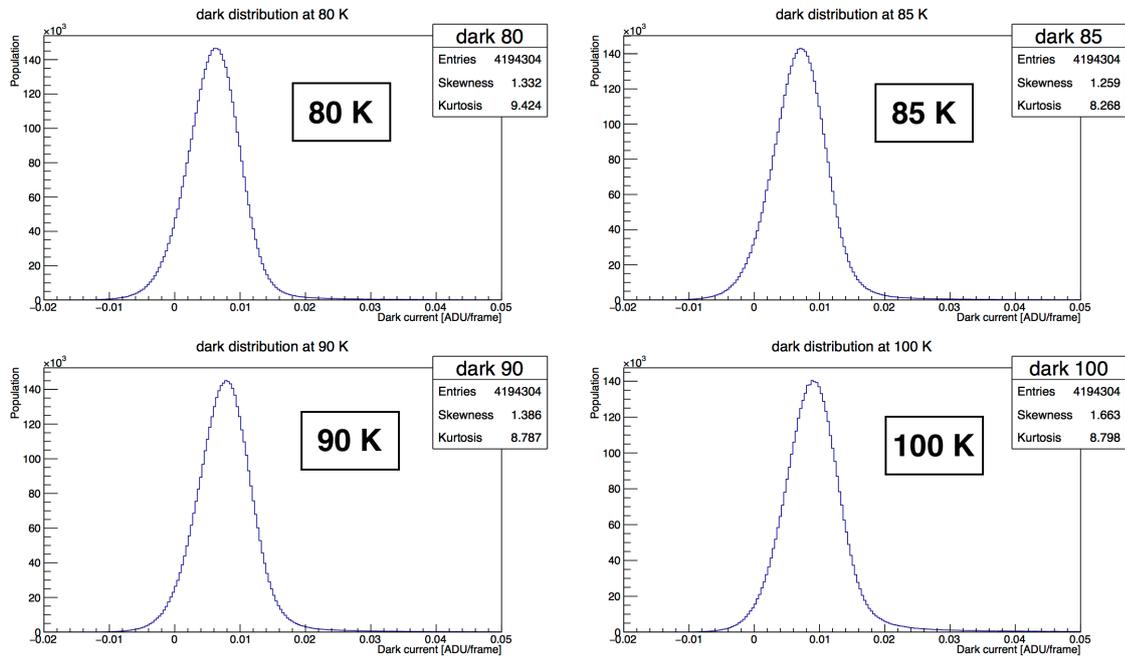


Figure 5.6: Dark current distributions for ASIC 2 at 80, 85, 90 and 100 K.

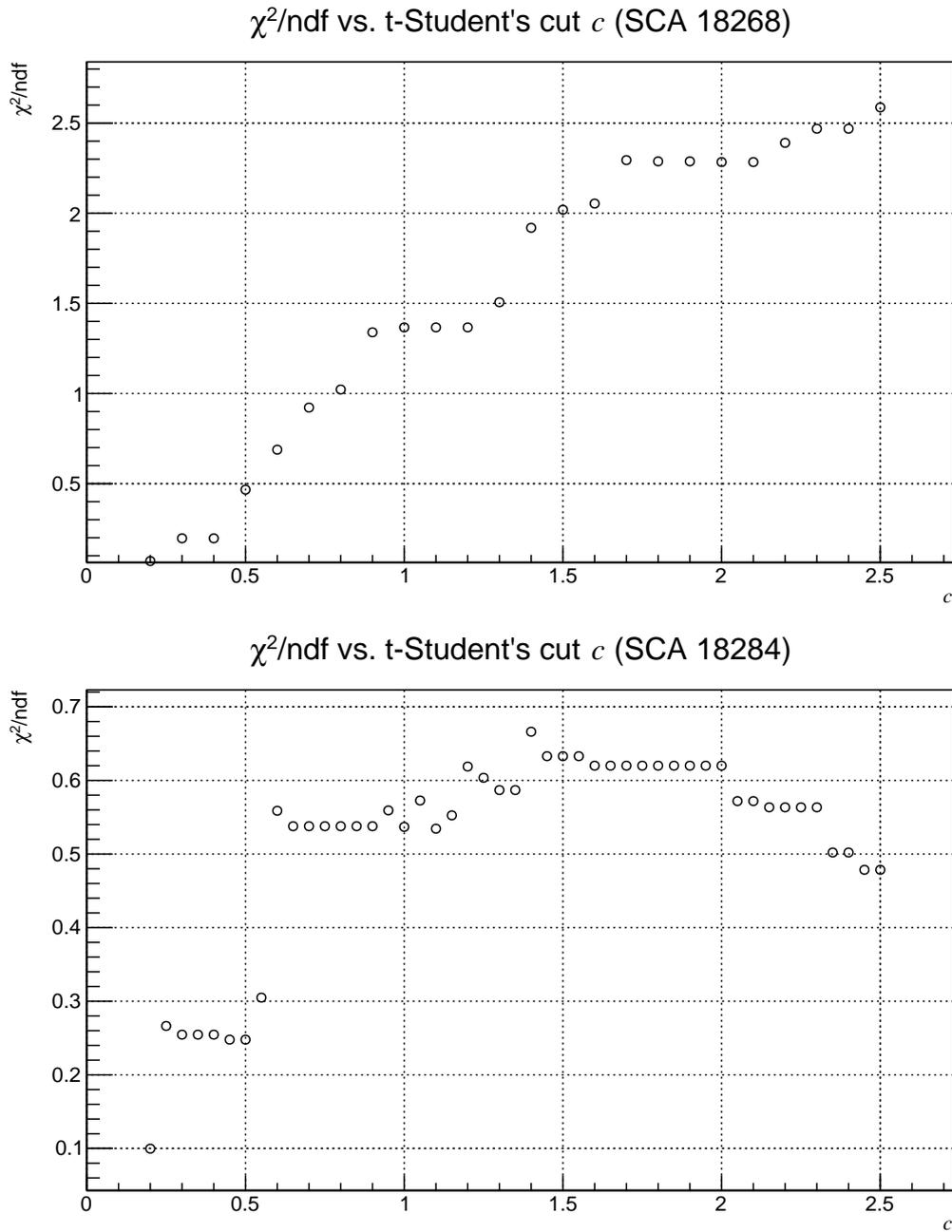


Figure 5.7:  $\chi^2/\text{ndf}$  of the Arrhenius fit for ASIC 1 (top panel) and ASIC 2 (bottom panel) versus  $c$  ( $c$  being the parameter defining the exclusion interval  $[\mu_t - c \cdot \sigma_t, \mu_t + c \cdot \sigma_t]$  in the  $t$ -Student's distribution).

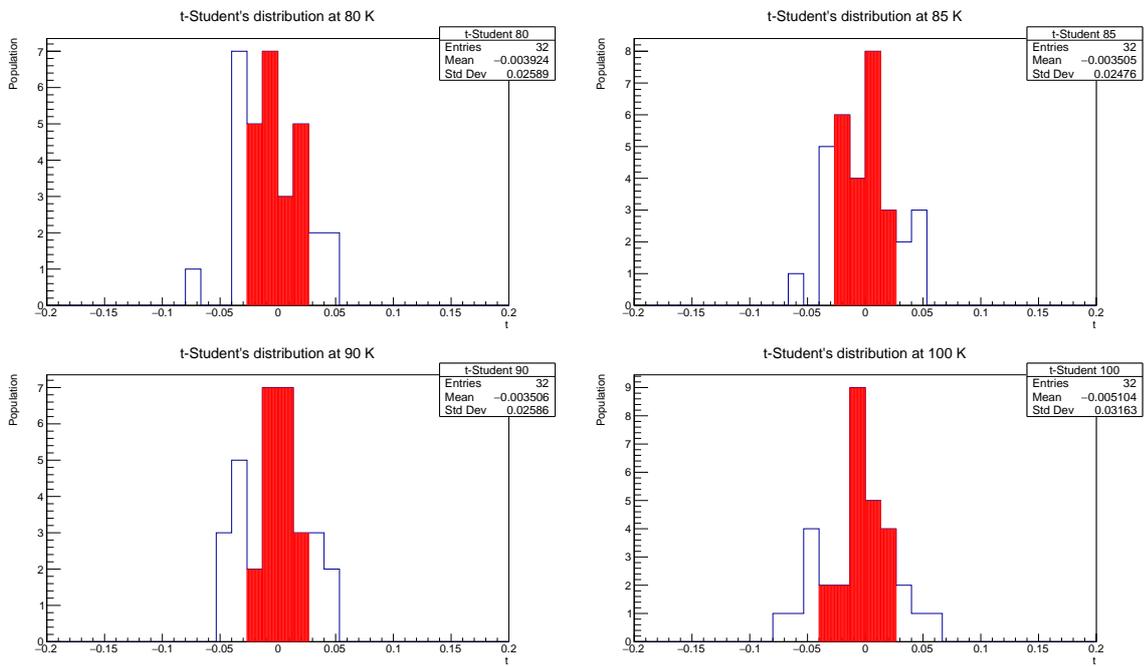


Figure 5.8:  $t$ -Student's distributions for ASIC 1. Red bins are within  $[\mu_t - 0.8\sigma_t, \mu_t + 0.8\sigma_t]$ .

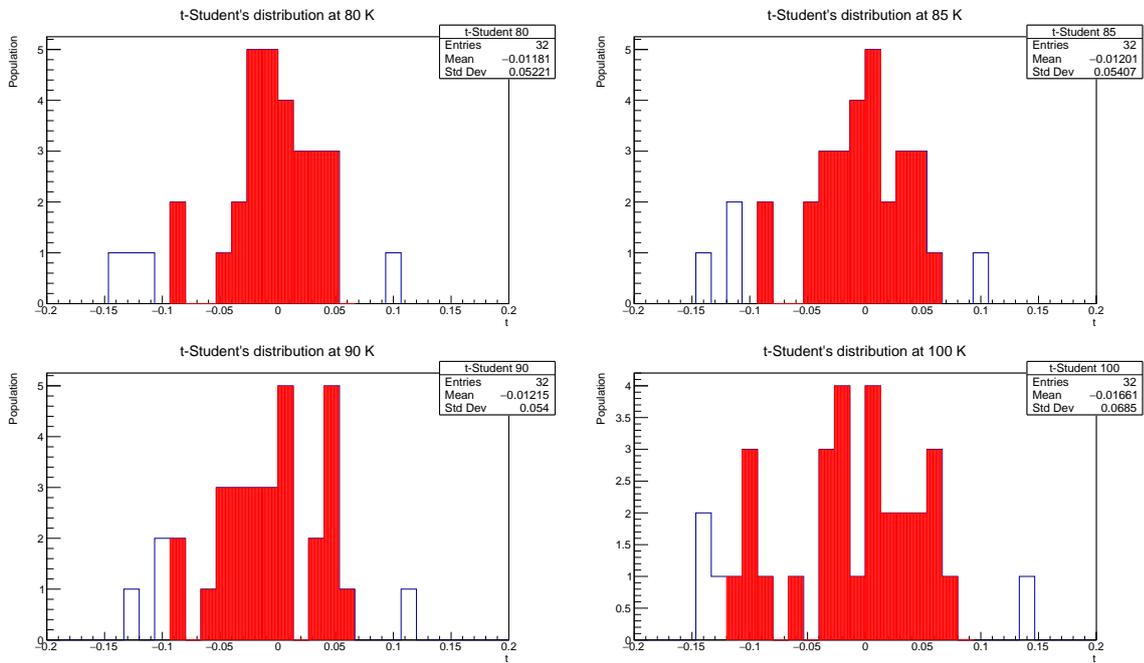


Figure 5.9:  $t$ -Student's distributions for ASIC 2. Red bins are within  $[\mu_t - 1.4\sigma_t, \mu_t + 1.4\sigma_t]$ .

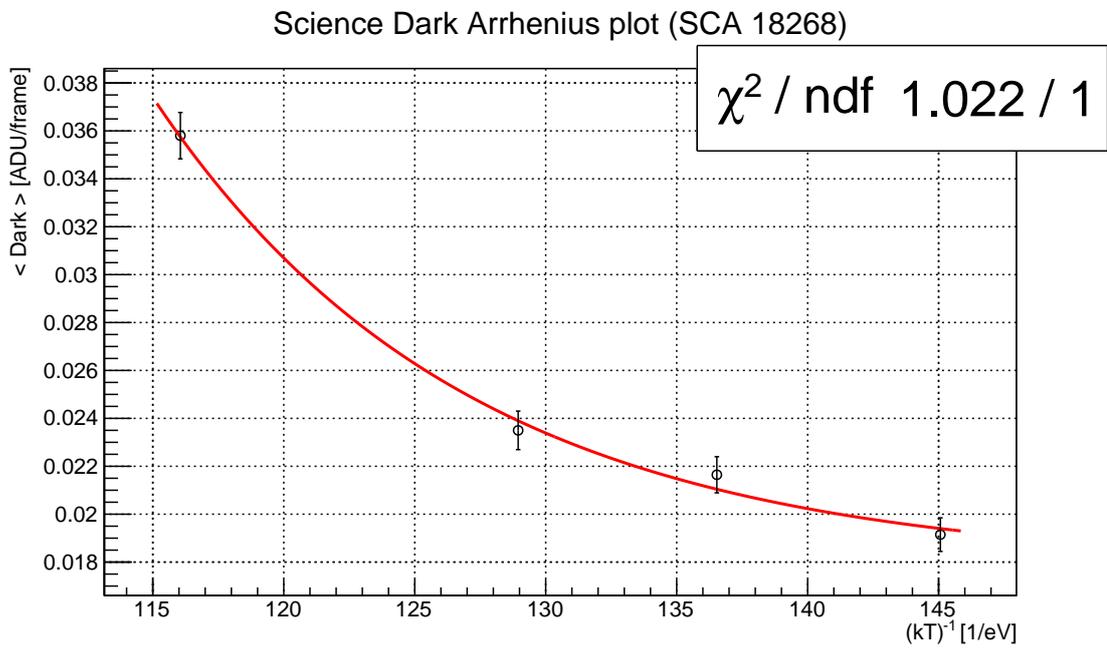


Figure 5.10: Arrhenius plot for ASIC 1 after selections on pixel channels.

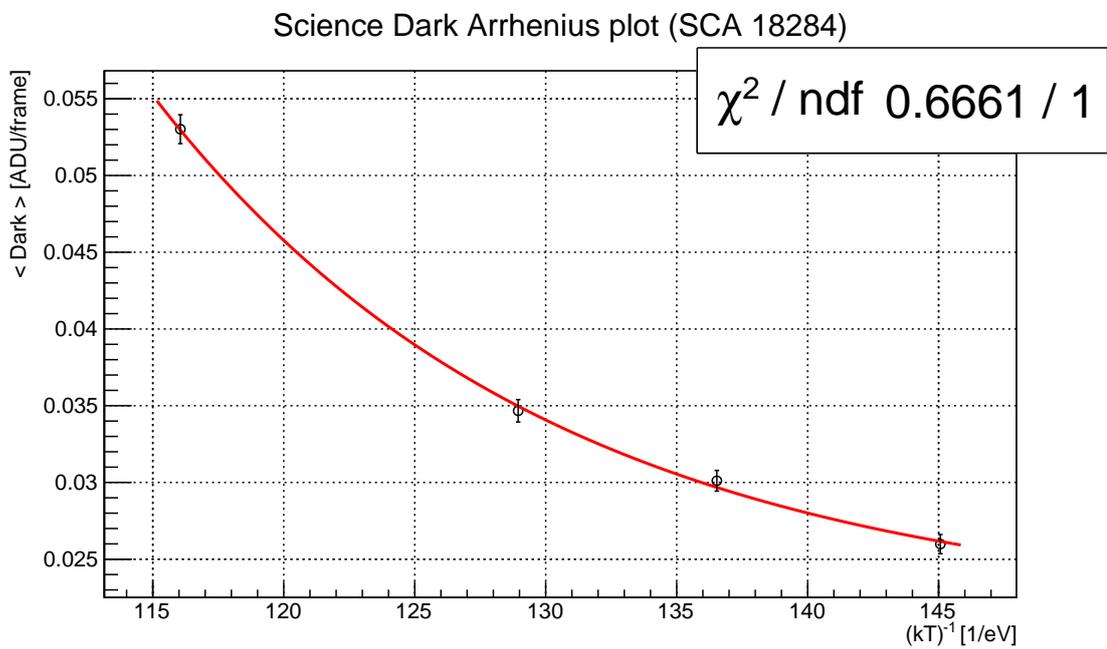


Figure 5.11: Arrhenius plot for ASIC 2 after selections on pixel channels.

	Temperature (K)	Skewness	Kurtosis
ASIC 1	80	1.6	14.1
	85	1.6	13.5
	90	1.8	14.9
	100	1.3	4.97
	Temperature (K)	Skewness	Kurtosis
ASIC 2	80	1.3	9.4
	85	1.3	8.3
	90	1.4	8.8
	100	1.7	8.8

Table 5.2: Table showing skewness and kurtosis of dark current distributions for ASIC 1 and ASIC 2.

### 5.3 Analysis results

The fit to the Arrhenius law yields a bandgap energy  $E_{bg,1} = (0.08 \pm 0.01)$  eV for ASIC 1 and  $E_{bg,2} = (0.07 \pm 0.01)$  eV for ASIC 2. These results are compatible with each other within uncertainties. The theoretical bandgap energy for HgCdTe crystals with cutoff wavelength  $\lambda_{cutoff} = 2.3 \mu\text{m}$  is greater by one order of magnitude ( $E_{bg} \sim 0.6$  eV, see Figure 4.12). However, Teledyne Imaging Systems reports that in the temperature range between 80 and 100 K, H2RG detectors with  $\sim 2.5 \mu\text{m}$  cutoff wavelength undergo a transition between different regimes. This transition is apparent in Figure 5.12 where the slope of the curve of dark current versus temperature changes rapidly. From our fit, a bandgap energy corresponding to  $\sim 13\%$  of the theoretical value was determined. This fraction is similar to the result obtained by James Webb Space Telescope collaboration for similar H2RG detectors ( $\lambda_{cutoff} = 5 \mu\text{m}$ ) in the 35-60 K temperature range [66],  $E_{bg} = 0.05$  eV. The value specified by Teledyne Imaging Systems is  $E_{bg} \sim 0.25$  eV (Figure 4.12): the observed bandgap energy corresponds to 20% of the theoretical value. Indeed, as shown in Figure 5.12, for  $\lambda_{cutoff} = 5 \mu\text{m}$  H2RG dark current in the temperature range between 35 and 60 K has not yet reached the nominal temperature dependence.

This analysis was performed to prepare the NISP instrument flight operation. The operation temperature of the detectors in flight will be kept at a fixed value between 80 and 100 K. If for any reasons the instrument has to operate at a different temperature than the operation one, it will be necessary to assess the detector dark current to continue the data processing. Measuring and modelling the dark current thermal dependency is then mandatory. The Arrhenius law fits to 40.6 % and 78.1 % of pixels for ASIC 1 and ASIC 2, respectively. The behaviour of ASIC 1 is less compatible with the Arrhenius law with respect to ASIC 2, but the maximum of  $\chi^2/ndf$  for ASIC 1 is  $\sim 2.5$ , as shown in Figure 5.7, that is almost within one standard deviation interval from 1. This kind of analysis has to be carried out also for H2RG detectors enclosed in *Andromède* cryostat, in order to compare the results. However, the current analysis shows that dark current dependence on temperature for *Pégase* H2RG detectors is well understood.

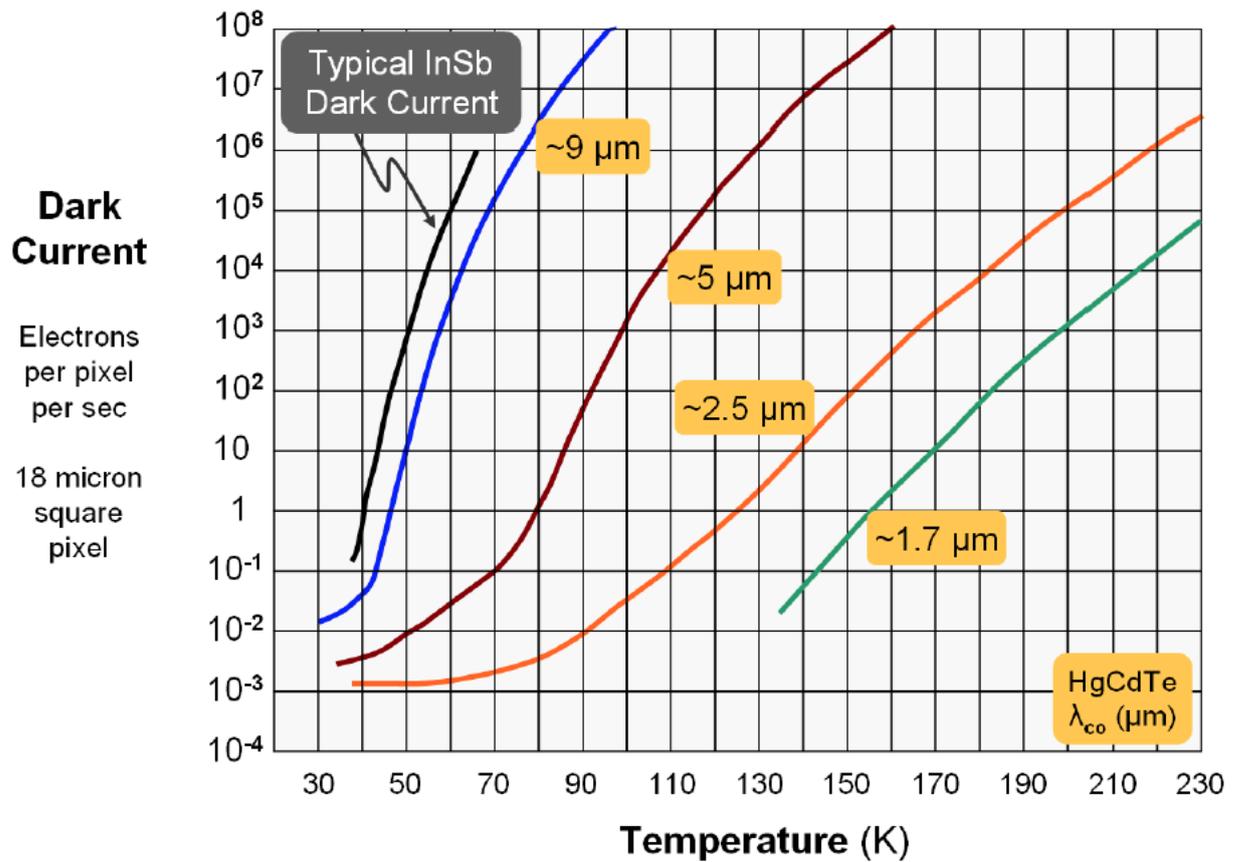


Figure 5.12: Dark current's dependence on temperature as a function of H2RG detector's cutoff wavelength [57].

The research reported in this thesis was carried out in the context of the ESA Euclid Mission in close collaboration with the Italian teams involved in the Euclid NISP instrument and with the French group at CPPM responsible for the NISP focal plane detector system. My contributions concerned in particular two aspects:

1. the design and implementation of a test environment aimed at developing and testing the communication between the Spacecraft and the NISP Instrument Control Unit;
2. the characterization of the NISP H2RG infrared detectors.

Regarding point 1, a software simulating the Spacecraft commanding and responding towards the NI-ICU was developed to allow the testing and validating of the functionalities of the ICU Application Software. The communication capability between our Spacecraft (S/C) simulator, the NISP Instrument Control Unit and Data Processing Unit preliminary models was successfully carried out, showing that the S/C simulator is able to accomplish a standard communication cycle for NISP observations. Moreover the S/C simulator is a useful tool for the development of the ICU Application Software and for the communication tests of NISP Warm Electronics.

Regarding point 2, the NISP focal plane detector is based on a matrix of  $2048 \times 2048$  pixels made of HgCdTe semiconductor crystal. I participated to the calibration campaign of the first available NISP H2RG detectors. I studied in detail the dark current of a couple of detectors at operation temperatures of 80-100 K. It was shown that the dependence of dark current on temperature is compatible with an exponential Arrhenius law with similar parameter values for both detectors. The estimated bandgap energy differs from the theoretical value, probably because H2RG detectors undergo a transition between different regimes at these temperatures. The ratio between the observed and theoretical bandgap energy is  $\sim 13\%$ : this result is compatible with that found by researchers working on James Webb Space Telescope (20%), equipped

with the same model of detectors.

The analysis results prove that H2RG dark current is compliant with Euclid requirements: the maximum average value is  $0.036 \text{ ADU/frame} = 0.05 \text{ e}^-/\text{s}$  for the first detector and  $0.053 \text{ ADU/frame} = 0.075 \text{ e}^-/\text{s}$  for the second one, to be compared with  $0.07 \text{ e}^-/\text{s}$  from mission specifications.

In conclusion, H2RG dark current behaviour in the Euclid operation temperature range can be considered well understood and ready to be integrated in the Euclid models.

In the upcoming months the characterization of 16 H2RG detectors, corresponding to the total number of those equipping the NISP focal plane, will be completed. By November 2018 the first readout electronics modules of the Flight Model will be characterized.

Looking forward, a first version of the NI-ICU Application Software, developed through the test equipment described in this thesis, is undergoing the integration phase in the NISP Avionic Model, which will be released to ESA by March 2018. Next version of the Application Software will be integrated in the NISP Qualification Model, including a part of the focal plane detectors. By September 2018 the first Thermal Vacuum tests of the NISP ICU-DPU Qualification Models is foreseen. Thermal Vacuum tests for performance verification and calibration of NISP subsystems will last until March 2019, and finally, by the end of March 2019, the NISP Flight Model will be delivered to ESA. If compliant with this schedule, the launch of the Euclid satellite will take place by the end of December 2020.

---

## Bibliography

---

- [1] Planck Collaboration, Ade, P. A. R., Aghanim, N., Arnaud, M., Ashdown, M., Aumont, J. et al., *Planck 2015 results - xiii. cosmological parameters*, *A&A* **594** (2016) A13.
- [2] R. Laureijs, J. Amiaux, S. Arduini, J. L. Auguères, J. Brinchmann, R. Cole et al., *Euclid Definition Study Report*, *ArXiv e-prints* (Oct., 2011) , [1110.3193].
- [3] A. Maeder, *An alternative to the  $\Lambda$ CDM model: The case of scale invariance*, *Astrophys. J.* **834** (2017) 194, [1701.03964].
- [4] A. Maeder, *Dynamical Effects of the Scale Invariance of the Empty Space: The Fall of Dark Matter?*, *apj* **849** (Nov., 2017) 158, [1710.11425].
- [5] A. Einstein, *Die Feldgleichungen der Gravitation*, *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften (Berlin)*, Seite 844-847. (1915) .
- [6] A. Friedmann, *Über die Krümmung des Raumes*, *Zeitschrift für Physik* **10** (1922) 377–386.
- [7] P. Coles and F. Lucchin, *Cosmology: The Origin and Evolution of Cosmic Structure*. John Wiley & Sons, 2002.
- [8] E. P. Hubble, *Extragalactic nebulae.*, *apj* **64** (Dec., 1926) .
- [9] F. Zwicky, *Die Rotverschiebung von extragalaktischen Nebeln*, *Helvetica Physica Acta* **6** (1933) 110–127.
- [10] V. C. Rubin and W. K. Ford, Jr., *Rotation of the Andromeda Nebula from a Spectroscopic Survey of Emission Regions*, *apj* **159** (Feb., 1970) 379.
- [11] P. J. E. Peebles, *Large-scale background temperature and mass fluctuations due to scale-invariant primeval perturbations*, *apjl* **263** (Dec., 1982) L1–L5.

- [12] J. R. Bond, A. S. Szalay and M. S. Turner, *Formation of galaxies in a gravitino-dominated universe*, *Phys. Rev. Lett.* **48** (Jun, 1982) 1636–1639.
- [13] G. R. Blumenthal, H. Pagels and J. R. Primack, *Galaxy formation by dissipationless particles heavier than neutrinos*, *nat* **299** (Sept., 1982) 37.
- [14] S. Perlmutter, G. Aldering, G. Goldhaber, R. A. Knop, P. Nugent, P. G. Castro et al., *Measurements of  $\Omega$  and  $\Lambda$  from 42 high-redshift supernovae*, *The Astrophysical Journal* **517** (1999) 565.
- [15] Astier, P., Guy, J., Regnault, N., Pain, R., Aubourg, E., Balam, D. et al., *The supernova legacy survey: measurement of  $\Omega_M$ ,  $\Omega_\Lambda$  and  $w$  from the first year data set*, *A&A* **447** (2006) 31–48.
- [16] BOOMERANG collaboration, P. de Bernardis et al., *A Flat universe from high resolution maps of the cosmic microwave background radiation*, *Nature* **404** (2000) 955–959, [astro-ph/0004404].
- [17] SDSS collaboration, M. Betoule et al., *Improved cosmological constraints from a joint analysis of the SDSS-II and SNLS supernova samples*, *Astron. Astrophys.* **568** (2014) A22, [1401.4064].
- [18] M. F. Barnothy and J. M. Barnothy, *Concentration of QSO's around  $z=2$  Redshift*, in *Bulletin of the American Astronomical Society*, vol. 1 of *baas*, p. 181, Mar., 1969.
- [19] W. L. Freedman, B. F. Madore, B. K. Gibson, L. Ferrarese, D. D. Kelson, S. Sakai et al., *Final results from the hubble space telescope key project to measure the hubble constant*, *The Astrophysical Journal* **553** (2001) 47.
- [20] R. A. Freedman and W. J. Kaufmann, *Universe*. W. H. Freedman and Company, New York, 2008.
- [21] PARTICLE DATA GROUP collaboration, C. Patrignani et al., *Review of Particle Physics*, *Chin. Phys.* **C40** (2016) 120.
- [22] C. G. Boehmer and T. Harko, *Physics of dark energy particles*, *Found. Phys.* **38** (2008) 216, [gr-qc/0602081].
- [23] E. Komatsu and D. N. Spergel, *Acoustic signatures in the primary microwave background bispectrum*, *Phys. Rev.* **D63** (2001) 063002, [astro-ph/0005036].
- [24] R. Durrer and R. Maartens, *Dark Energy and Modified Gravity*, in *Dark Energy: Observational & Theoretical Approaches*, ed. P. Ruiz-Lapuente (Cambridge UP, 2010), pp48 - 91, pp. 48 – 91, 2008, 0811.4132, <https://inspirehep.net/record/803511/files/arXiv:0811.4132.pdf>.
- [25] P. J. E. Peebles, *The large-scale structure of the universe*. Princeton University Press, 1980.

- [26] R. Massey et al., *COSMOS: 3D weak lensing and the growth of structure*, *Astrophys. J. Suppl.* **172** (2007) 239–253, [astro-ph/0701480].
- [27] A. H. Guth, *Inflationary universe: A possible solution to the horizon and flatness problems*, *Phys. Rev. D* **23** (Jan, 1981) 347–356.
- [28] B. R. Greene, *The fabric of the cosmos : space, time, and the texture of reality*. Penguin, London, 2004.
- [29] D. J. Eisenstein, I. Zehavi, D. W. Hogg, R. Scoccimarro, M. R. Blanton, R. C. Nichol et al., *Detection of the baryon acoustic peak in the large-scale correlation function of sdss luminous red galaxies*, *The Astrophysical Journal* **633** (2005) 560.
- [30] A. S. Szalay, *The large scale structure of the universe*, *AIP Conference Proceedings* **207** (2017/10/30, 1990) 257–263.
- [31] Racca D. Giuseppe et al., *The Euclid mission design*, *Proc. SPIE* **9904** (2016) 9904 – 9904 – 23.
- [32] J. Amiaux et al., *Euclid mission: building of a reference survey*, *Proc. SPIE* **8442** (2012) 8442 – 8442 – 11.
- [33] R. Bogiatto, *Euclid Spacecraft Communication Interface Control Document*, *EUCL-TAST-ICD-1-004* (2016) .
- [34] D. Korsch, *Closed form solution for three-mirror telescopes, corrected for spherical aberration, coma, astigmatism, and field curvature*, *Appl. Opt.* **11** (Dec, 1972) 2986–2987.
- [35] P. Gare, G. Sarri and R. Schmidt, “ESA’s ‘Billion-Pixel’ Camera.” <http://www.esa.int/esapub/bulletin/bulletin137/bulletin137.pdf>, February 2009.
- [36] M. Cropper et al., *VIS: the visible imager for Euclid*, *Proc. SPIE* **9904** (2016) 9904 – 9904 – 16.
- [37] T. Maciaszek et al., *Euclid Near Infrared Spectrometer and Photometer instrument concept and first test results obtained for different breadboards models at the end of phase C*, *Proc. SPIE* **9904** (2016) .
- [38] S. Ligorì et al., *Detailed design and first tests of the application software for the instrument control unit of Euclid-NISP*, *Proc. SPIE* **9904** (2016) 9904 – 9904 – 12.
- [39] “EDISOFT Defence & Aerospace Technologies.” <http://http://www.edisoft.pt>.
- [40] L. Valenziano, E. Maiorano, M. Trifoglio, C. Bonoli, F. Bortoletto, S. Ligorì et al., *NISP Instrument Operation Concept Document*, *EUCL-IBO-PL-7-001* (2017) .
- [41] “GRMON by Cobham Gaisler.” <http://www.gaisler.com/index.php/products/debug-tools/grmon>.

- [42] “Actel Flash-Pro4 by Microsemi Corporation.” <https://www.microsemi.com/products/fpga-soc/design-resources/programming/flashpro>.
- [43] “MIL-STD-1553 Tutorial.” <http://www.geembedded.com>, 2015.
- [44] “USB 1553 Avionics Interface by Astronics Ballard Technology.” <https://www.astronics.com/ballard-technology/test-simulation-interfaces/usb-1553-usb-avionics-interfaces-for-mil-std-1553>.
- [45] R. Bogiatto and C. Maddaleno, *MIL-STD-1553B System Data Bus Protocol Specification, EUCL-TAST-RS-1-017* (2016) .
- [46] F. Bortoletto, C. Bonoli and M. D’Alessandro, *NI-DPU Commanding and HouseKeeping Tables, EUCL-OPD-TN-7-004* (2017) .
- [47] “VxWorks by Wind River Systems.” <https://www.windriver.com/products/vxworks/>.
- [48] J. W. Beletic and M. Loose, *Scientific imaging sensors, Presented at the workshop ‘Detectors for astronomy’* (2009) .
- [49] E. Fossum, *Electron devices, IEEE Transactions on* **44** (10) (1997) 1689–1698.
- [50] A. C. Moore, Z. Ninkov and W. J. Forrest, *Interpixel capacitance in nondestructive focal plane arrays*, in *Focal Plane Arrays for Space Telescopes* (T. J. Grycewicz and C. R. McCreight, eds.), vol. 5167 of *procspie*, pp. 204–215, Jan., 2004, DOI.
- [51] R. P. Dudik, M. E. Jordan, B. N. Dorland, D. Veillette, A. Waczynski, B. F. Lane et al., *Interpixel crosstalk in Teledyne Imaging Sensors H4RG-10 detectors, Appl. Opt.* **51** (May, 2012) 2877–2887.
- [52] W. Shockley and W. T. Read, *Statistics of the recombinations of holes and electrons, Phys. Rev.* **87** (Sep, 1952) 835–842.
- [53] R. N. Hall, *Electron-hole recombination in germanium, Phys. Rev.* **87** (Jul, 1952) 387–387.
- [54] J. D. Vincent, *Fundamentals of infrared and visible detector operation and testing*. John Wiley & Sons, 2015.
- [55] G. L. Hansen, J. L. Schmit and T. N. Casselman, *Energy gap versus alloy composition and temperature in  $Hg_{1-x}Cd_xTe$ , Journal of Applied Physics* **53** (1982) 7099–7101, [<https://doi.org/10.1063/1.330018>].
- [56] R. Barbier, *NI-SCS characterisation requirements, EUCL-CPP-TN-7-004* (2015) .
- [57] J. W. Beletic et al., *Teledyne imaging sensors: infrared imaging technologies for astronomy and civil space, Proc. SPIE* **7021** (2008) 7021 – 7021 – 14.

- [58] L. Dressel, *Wide field camera 3 instrument handbook, version 9.0*, Baltimore: STScI (2017) .
- [59] M. Loose et al., *The SIDECAR ASIC: focal plane electronics on a single chip*, *Proc. SPIE* **5904** (2005) 5904 – 5904 – 10.
- [60] F. Bortoletto, *Warm electronics to focal plane ICD*, *EUCL-OPD-ICD-7-002* (2014) .
- [61] A. Rogalski, *HgCdTe infrared detector material: history, status and outlook*, *Reports on Progress in Physics* **68** (2005) 2267.
- [62] J. Janesick, *CCD characterization using the photon transfer technique*, *Proc. SPIE* **570** (1985) 7–19.
- [63] B. Kubik, *Reference pixel subtraction method for EUCLID SCS noise specification*, *EUCL-IPN-TN-7-001* (2014) .
- [64] “Acktar Fractal Black.” <http://www.acktar.com/category/BlackOpticalCoating>.
- [65] “The HDF5 Group (1997-2017).” <https://support.hdfgroup.org/HDF5/>.
- [66] B. J. Rauscher et al., *The Dark Current and Hot Pixel Percentage of James Webb Space Telescope 5  $\mu$ m Cutoff HgCdTe Detector Arrays as Functions of Temperature*, *Publications of the Astronomical Society of the Pacific* **123** (2011) 953–957.
- [67] R. N. Sajjad, W. Chern, J. L. Hoyt and D. A. Antoniadis, *Trap Assisted Tunneling and Its Effect on Subthreshold Swing of Tunnel FETs*, *IEEE Transactions on Electron Devices* **63** (Nov., 2016) 4380–4387, [1603.06654].

# Appendices

<b>A</b>	<b>Acronym List</b>	<b>154</b>
<b>B</b>	<b>ICU-ASW building, running and debugging</b>	<b>158</b>
B.1	Building and running the ICU-ASW . . . . .	158
B.1.1	<i>Eclipse</i> configuration . . . . .	158
B.1.2	Downloading the ICU-ASW source code . . . . .	160
B.1.3	Building <i>RTEMS</i> . . . . .	160
B.1.4	Creating an <i>Eclipse</i> project to build the ICU-ASW . . . . .	161
B.1.5	Loading and running the ICU-ASW on the <i>EBB</i> with <i>GRMON</i> . . . . .	171
B.1.6	Running the ICU-ASW in <i>Debug mode</i> with <i>Eclipse</i> . . . . .	174
B.2	Building and running <i>Edisoft RTEMS</i> tests . . . . .	179
B.2.1	Building and running the <i>testsuite</i> . . . . .	179

## APPENDIX A

# Acronym List

Acronym	Meaning
ADC	Analog-to-Digital Converter
ADU	Analog-to-Digital Unit
AOCS	Attitude and Orbit Control System
APID	Application Process ID
ASIC	Application Specific Integrated Circuit
ASW	Application SoftWare
ATC	Acquisition Transfer Confirmation
ATR	Acquisition Transfer Request
BAO	Baryonic Acoustic Oscillation
BC	Bus Controller
BM	Bus Monitor
BSP	Board Support Package
BSW	Boot SoftWare
CAD	Computer Aided Design
CCD	Charge-Coupled Device
CDM	Cold Dark Matter
CDMS	Command and Data Management System
CDMU	Command and Data Management Unit
CDPU	Control and Data Processing Unit
CDS	Correlated Double Sampling
CERN	Centre Européen pour la Recherche Nucléaire
CFRP	Carbon Fiber Reinforced Plastic
CMB	Cosmic Microwave Background

<b>Acronym</b>	<b>Meaning</b>
CMOS	Complementary Metal Oxide Silicon
CPPM	Centre de Physique des Particules de Marseille
CPU	Central Processing Unit
CU	Calibration Unit
DC	Direct Current
DCU	Detector Control Unit
DE	Dark Energy
DM	Dark Matter
DPU	Data Processing Unit
DS	Detector System
DTC	Distribution Transfer Confirmation
DTD	Distribution Transfer Descriptor
ECSS	European Cooperation for Space Standardization
EGSE	Electrical Ground Support Equipment
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESA	European Space Agency
FDIR	Failure Detection, Identification and Recovery
FGS	Fine Guidance Sensor
FITS	Flexible Image Transport System
FM 1, FM 2, FM 3	Folding Mirror 1, 2, 3
FOM	Figure Of Merit
FOV	Field Of View
FPA	Focal Plane Array
FPGA	Field Programmable Gate Array
FRW	Fridman-Robertson-Walker
FWA	Filter Wheel Assembly
FWHM	Full Width Half Maximum
GC	Galaxy Clustering
GR	General Relativity
GWA	Grism Wheel Assembly
H2RG	HAWAII 2RG - HgCdTe Astronomy Wide Area Infrared Imager (2k) <sup>2</sup> Resolution, Reference pixels and Guide mode
HDM	Hot Dark Matter
HK	HouseKeeping
HGA	High-Gain Antenna
HgCdTe	Hg (Mercury) Cadmium Telluride
HST	Hubble Space Telescope
HW	HardWare
ICD	Interface Control Document
ICU	Instrument Control Unit
INAF	Istituto Nazionale di AstroFisica
IR	InfraRed
ISW	Integrated Sachs-Wolfe effect
JWST	James Webb Space Telescope
L0, L1	Level 0, Level 1 Euclid data
L2	Second Lagrange Point

<b>Acronym</b>	<b>Meaning</b>
ACDM	Lambda Cold Dark Matter
LED	Light Emitting Diode
LGA	Low-Gain Antenna
LHC	Large Hadron Collider
LSF	Least Square Fit
LVDS	Low Voltage Differential Signal
M1, M2, M3	primary, secondary, tertiary Mirrors
M2M	M2 Mechanism (subsystem)
MACC	Multi Accumulation Readout
MIL-STD-1553	United States Department of Defense Military Standard 1553
MIPS	Mega Instructions Per Second
MLI	Multi Layer Insulation
MMU	Mass-Memory Unit
Mpc	Megaparsec
NASA	National Aeronautic and Space Administration
NI-ASW	NISP Application SoftWare
NI-CU	NISP Calibration Unit
NI-DCU	NISP Detector Control Unit
NI-DPU	NISP Data Processing Unit
NI-DS	NISP Detector System
NI-FPA	NISP Focal Plane Assembly
NI-FWA	NISP Filter Wheel Assembly
NI-GWA	NISP Grism Wheel Assembly
NI-ICU	NISP Instrument Control Unit
NI-OMA	NISP Opto-Mechanical Assembly
NIR	Near InfraRed
NI-SCS	NISP Sensor Chip System
NISP	Near Infrared Spectrometer and Photometer
NI-WE	NISP Warm Electronics
PCB	Printed Circuit Board
PCDU	Power Conditioning and Distribution Unit
PLM	PayLoad Module
PMCU	Power and Mechanisms Control Unit
PSF	Point Spread Function
PSU	Power Supply Unit
PTC	Photon Transfer Curve
QC	Quality Check
QE	Quantum Efficiency
RMS	Root Mean Square
ROIC	ReadOut Integrated Circuit
RSD	Redshift-Space Distortion
RT	Remote Terminal
S/C	SpaceCraft
SCA	Sensor Chip Assembly
SCE	Sensor Chip Electronics
SCS	Sensor Chip System

<b>Acronym</b>	<b>Meaning</b>
SDSS	Sloan Digital Sky Survey
SGS	Science Ground Segment
SiC	Silicon Carbide
SIDECAR	System for Image Digitization, Enhancement, Control And Retrieval
SN	SuperNova
SN Ia	Type Ia SuperNova
SNR	Signal-to-Noise Ratio
SU	Shutter Unit
SVM	SerVice Module
STScI	Space Telescope Science Institute
SW	Software
SWL	SpaceWire Link
TC	TeleCommands
TIS	Teledyne Imaging System
TM	TeleMetry
TN	Technical Note
TT&C	Telemetry, Tracking and Command
TV	Thermal Vacuum
UAH	University of Alcalá de Henares
UTR	Up The Ramp readout mode
VI-CU	VIS Calibration Unit
VI-FPA	VIS Focal Plane Assembly
VIS	VISible Instrument
VI-RSU	VIS Read-out Shutter Unit
VI-SU	VIS Shutter Unit
WE	Warm Electronics
WIMP	Weakly Interacting Massive Particle
WL	Weak Lensing
WMAP	Wilkinson Microwave-Anisotropy Probe

## APPENDIX B

# ICU-ASW building, running and debugging

This document describes the procedure to be followed in order to compile the *NISP Instrument Control Unit Application SoftWare* (ICU-ASW) on Linux platform. Besides, an explanation of how to load and run the ICU-ASW on the *EBB* breadboard through *GRMON software* is reported. Finally, the instructions to compile and run the *RTEMS operating system* validation tests (provided by *Edisoft*) are described.

## B.1 Building and running the ICU-ASW

A 64-bit PC with Ubuntu 16.04 LTS operating system was used by Bologna INFN team. The PC-EBB interface is made with a serial link RS-232 communicating with the *Debug Support Unit (DSU)* of the *LEON2* processor through *GRMON software*.

### B.1.1 *Eclipse* configuration

The development environment used to write and compile the ICU-ASW is the one provided by *Eclipse* with *C/C++ Development Toolkit*<sup>1</sup>. *Eclipse* requires the installation of *Java Runtime Environment* (latest version can be downloaded at <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>). Once *JRE* is installed, *Eclipse* must be provided with a series of plugins in order to compile *RTEMS* operating system applications, to produce related documentation and to debug executables loaded on *EBB LEON2* processor via *GRMON*. These plugins can be installed in the following way:

- start *Eclipse*: at first run it will ask to select a directory to be used as *workspace*, then this request can be disabled selecting the option **Use this as the default and do not ask again**;
- open the *Workbench* (also this operation will be required only at first run, then it will be executed automatically);

---

<sup>1</sup>The currently used version can be downloaded at [http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/juno/SR2/eclipse-cpp-juno-SR2-linux-gtk-x86\\_64.tar.gz](http://www.eclipse.org/downloads/download.php?file=/technology/epp/downloads/release/juno/SR2/eclipse-cpp-juno-SR2-linux-gtk-x86_64.tar.gz)

- on the menu bar select **Help** (first from the right) and on the drop down menu select **Install New Software**; the window in Figure B.1 will open and then the url of the websites where the plugins can be downloaded must be provided at **Work with:**;
- *CDT (C/C++ Development Toolkit)* plugins can be installed clicking on **Add...**, just at the right of **Work with:**; a window will open: at **Name**, `cdt` must be inserted, while at **Location** the address `http://download.eclipse.org/tools/cdt/releases/8.8.1` must be provided, then click on **OK**; select **CDT Main Features** and **CDT Optional Features** and then click on **Next >**; if an incompatibility message appears, deselect indicated packages and retry the installation; when all incompatible packages are deselected, it will be possible to click on **Finish** and the installation will end;

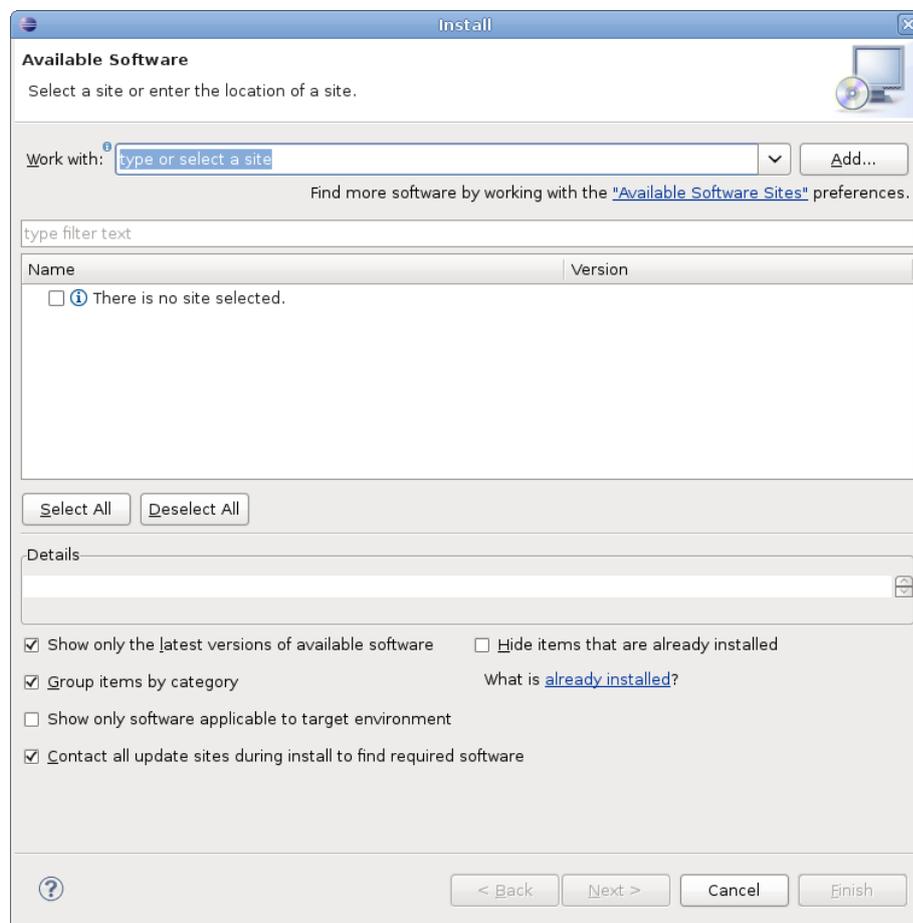


Figure B.1: *The Eclipse window to install plugins.*

- *Doxygen* plugins can be installed clicking on **Add...**; a window will open: at **Name**, `doxygen` must be inserted, while at **Location** the address `http://download.gna.org/eclox/update` must be provided, then click on **OK**; select `eclox` and then click on **Next**

>; no incompatibility messages should appear in principle, so that it will be possible to click on **Finish** and to complete the installation;

- *GRMON* plugins can be installed clicking on **Add...**; a window will open: at **Name**, **grmon** must be inserted, while at **Location** the address [http://gaisler.com/eclipse/lide\\_kepler](http://gaisler.com/eclipse/lide_kepler) must be provided, then click on **OK**; select **LEON IDE** and then click on **Next** >; no incompatibility messages should appear in principle, so that it will be possible to click on **Finish** and to complete the installation;
- *RTEMS* plugins can be installed clicking on **Add...**; a window will open: at **Name**, **rtems** must be inserted, while at **Location** the address <ftp://ftp.rtems.org/pub/rtems/eclipse/updates> must be provided, then click on **OK**; select **ALL** and then click on **Next** >; no incompatibility messages should appear in principle, so that it will be possible to click on **Finish** and to complete the installation.

### B.1.2 Downloading the ICU-ASW source code

The ICU-ASW source code can be downloaded as compressed file from the *Git* repository at <https://baltig.infn.it/euclid/ICU-ASW/commits/master>, where a commit can be selected. Here, click on top right at **Browse Files** and then again on top right click on the symbol



: a drop down menu will open in order to select the compression file format (**zip**, **tar**, **tar.gz** or **tar.bz2**). Alternatively, the remote *Git* repository at <https://baltig.infn.it/euclid/ICU-ASW> can be cloned in a local repository. A useful GUI interface to manage *Git* directories on Linux operating system is *GitKraken* (<https://www.gitkraken.com>).

### B.1.3 Building *RTEMS*

*RTEMS 4.8 (tailored version)* source code is available on the CD provided by *Edisoft*. To build *RTEMS*, two procedures can be followed:

- running a *Debian 5* virtual machine through *Virtualbox*; *RTEMS* can be compiled following the instructions reported on the *User Manual* provided by *Edisoft* on the same CD;
- running a *Docker container* based on a *Docker image*, which can be produced downloading the files and following the instructions reported at the *Git* repository <https://baltig.infn.it/euclid/docker>; to run the container it is sufficient to install *Docker* (<https://www.docker.com>), load the *Docker image* which should be named **icu-devel.tgz** with the command `docker load < /path/to/icu-devel.tgz` and then launch the command `docker run -it --rm icu-devel /bin/bash`; to share a directory between the container and the host operating system it is sufficient to include the **-v** option: for instance, to share the directory **\$HOME/WORKSPACE** of the host operating system, the command `docker run -it --rm -v $HOME/WORKSPACE:$HOME/WORKSPACE icu-devel /bin/bash` must be run.

To compile the ICU-ASW source code, some environment variables must be set, modifying the `.bashrc` file. Some variables must be set according to the instructions provided by *Edisoft* User Manual, but concerning *Edilib*, `/opt` is suggested to be used as mounting directory instead of `/tmp` (section 4.2.1 of the User Manual):

- `export RIMP_ROOT = /home/rtems;`
- `export PATH = $RIMP_ROOT/rtems/bin:${PATH};`
- `export RTEMS_EXTERNAL_LIBRARY_PATH = /opt/edilib/library;`
- `export BSP = ebb;`
- `export RTEMS_MAKEFILE_PATH = $RIMP_ROOT/rtems/sparc-rtems4.8/$BSP;`
- `export PROJECT_ROOT = $RIMP_ROOT/build/rtems-impr.`

Before configuring and building *RTEMS* (section 4.3.3 of the User Manual), it is necessary to bootstrap the EBB *Board Support Package*. The following procedure must be executed:

- copy the `ebb` directory at the *Git* repository at <https://baltig.infn.it/euclid/rtems-impr/tree/ebb/c/src/lib/libbsp/sparc> to the path `$PROJECT_ROOT/c/src/lib/libbsp/sparc;`
- copy the `ebb.cfg` file at <https://baltig.infn.it/euclid/rtems-impr/tree/ebb/make/custom> to the path `$PROJECT_ROOT/make/custom;`
- in the `$PROJECT_ROOT` folder run the command `./bootstrap -c`, then `./bootstrap -p` and finally `./bootstrap`.

#### B.1.4 Creating an *Eclipse* project to build the ICU-ASW

Once *RTEMS* is built, the ICU-ASW can be compiled on Ubuntu 16.04 by copying the folder at `$RIMP_ROOT` from *Debian 5* virtual machine or *Docker container* to host operating system. The same procedure must be repeated for `/opt/edilib` folder. Moreover, the *RTEMS* environment variables set in the previous section must be inserted also in the `.bashrc` file of the host operating system. To share a folder between the host and the *Docker container* we have already mentioned the `-v` option, whereas for a *Virtualbox* virtual machine the same task can be accomplished clicking on **Settings** from the menu bar, and then on **Shared Folders** → **Add New Shared Folder**. Here, a folder of the host operating system must be inserted. When the virtual machine is running, the folder previously inserted can be shared in the virtual machine via the command `mount.vboxsf host_folder path/to/virtual_machine_folder`. Now, the *Eclipse* project to compile the ICU-ASW source code must be created. On the menu bar select **File** → **New Project** → **C Project**. Then name the project as ICU-ASW, select **Executable** → **Empty Project** as project type and **Cross GCC** as *Toolchain* (Figure B.2). Click on **Next** > and the window in Figure B.3 will appear. Then click again on **Next** > and select *sparc C* compiler as **Cross GCC Command** (Figure B.4). Finally click on **Finish** and the ICU-ASW project is actually created. The project should be visible in the *Project Explorer* panel of the

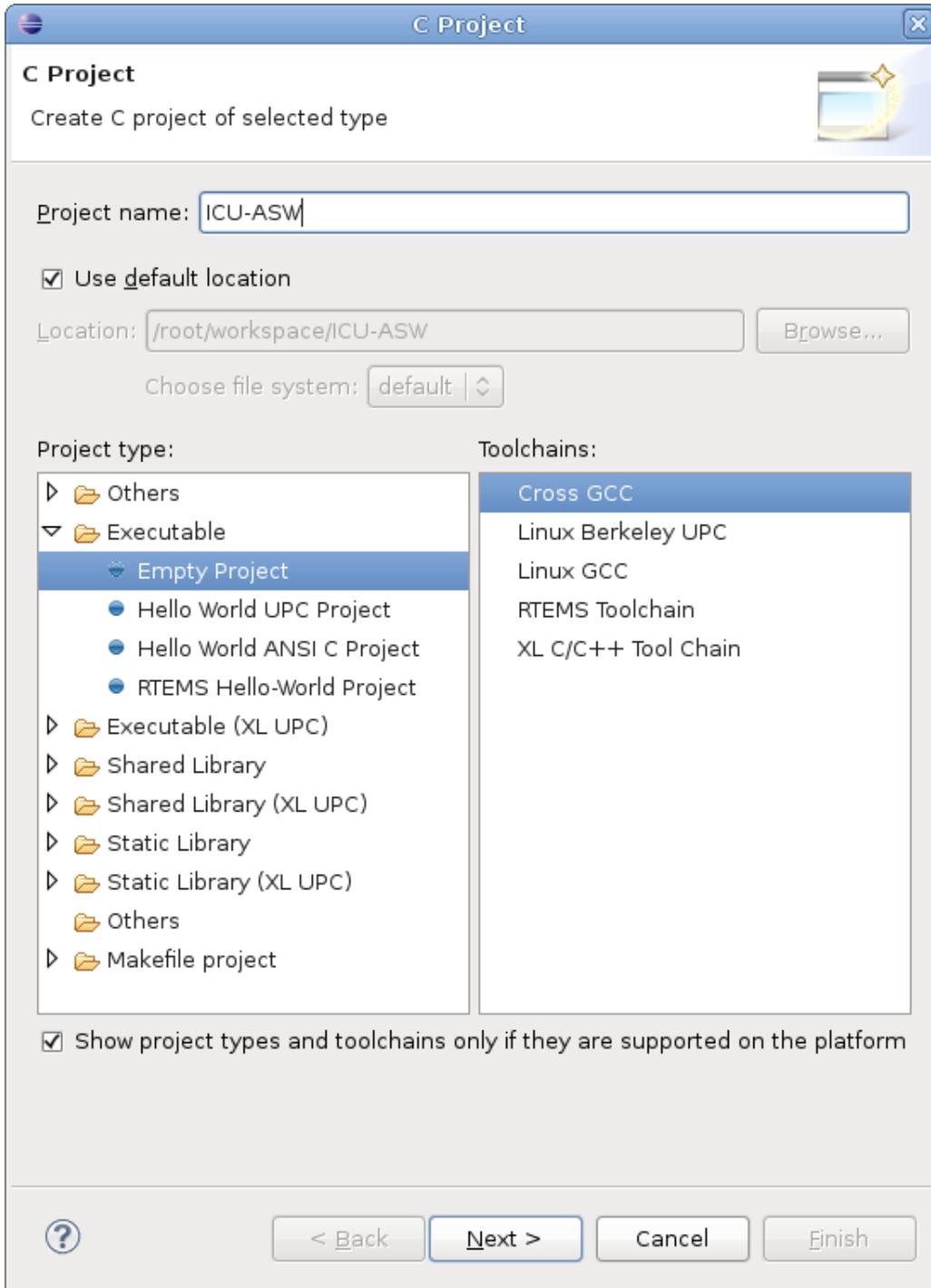


Figure B.2: Select the Toolchain for the Eclipse C project.

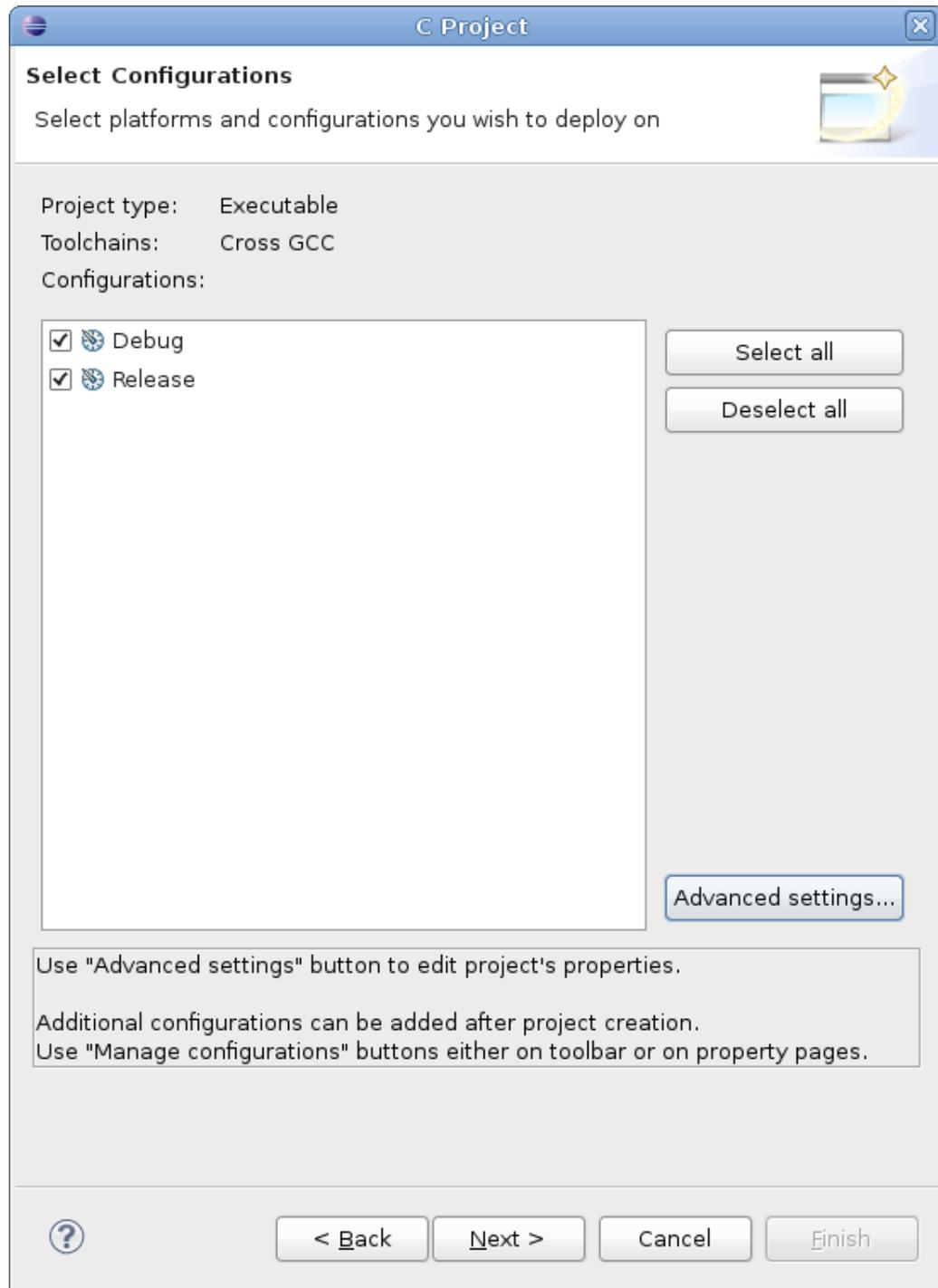


Figure B.3: *Select the configuration for the Eclipse C project.*

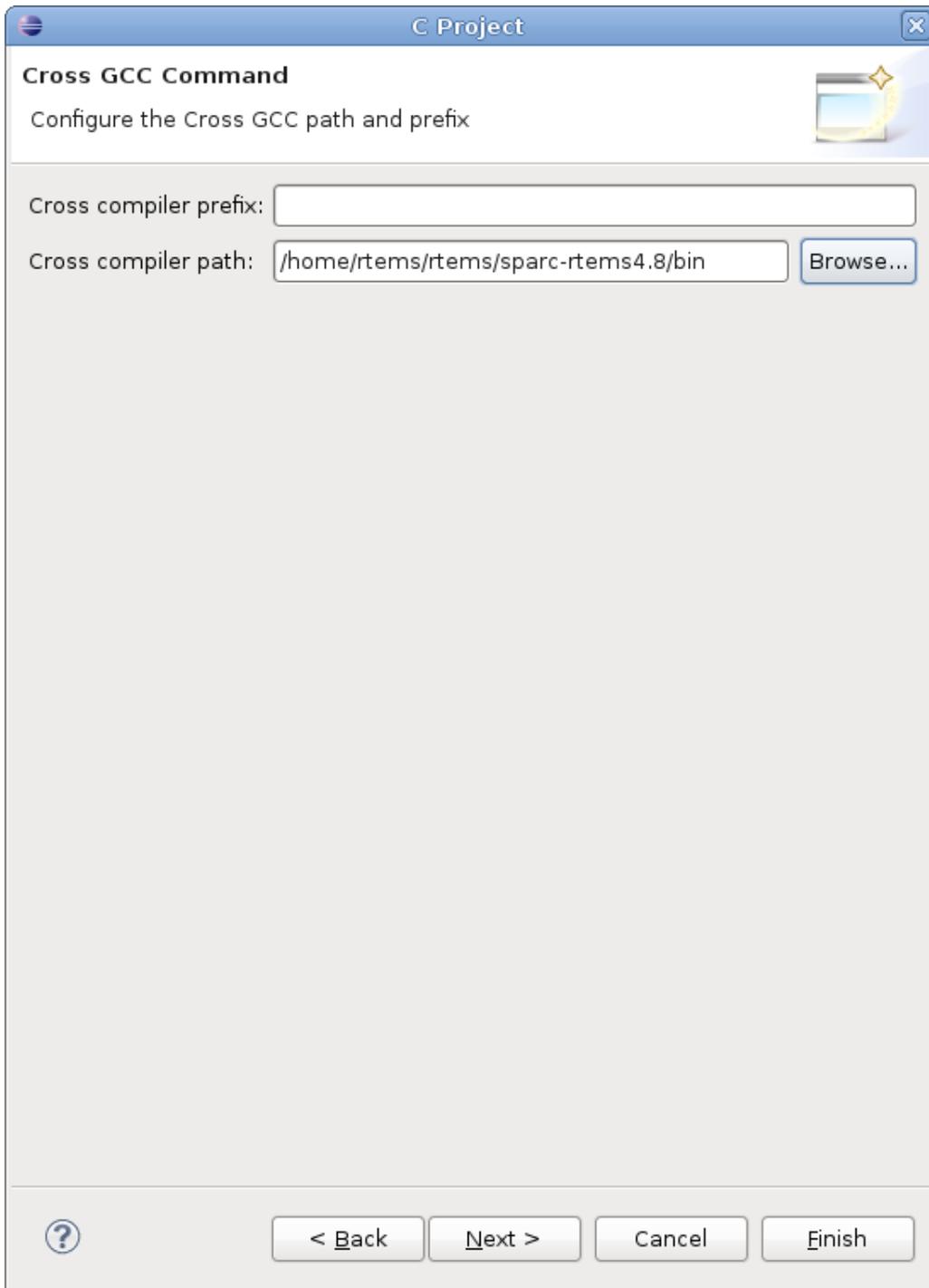


Figure B.4: Select the cross compiler path for the Eclipse C project.

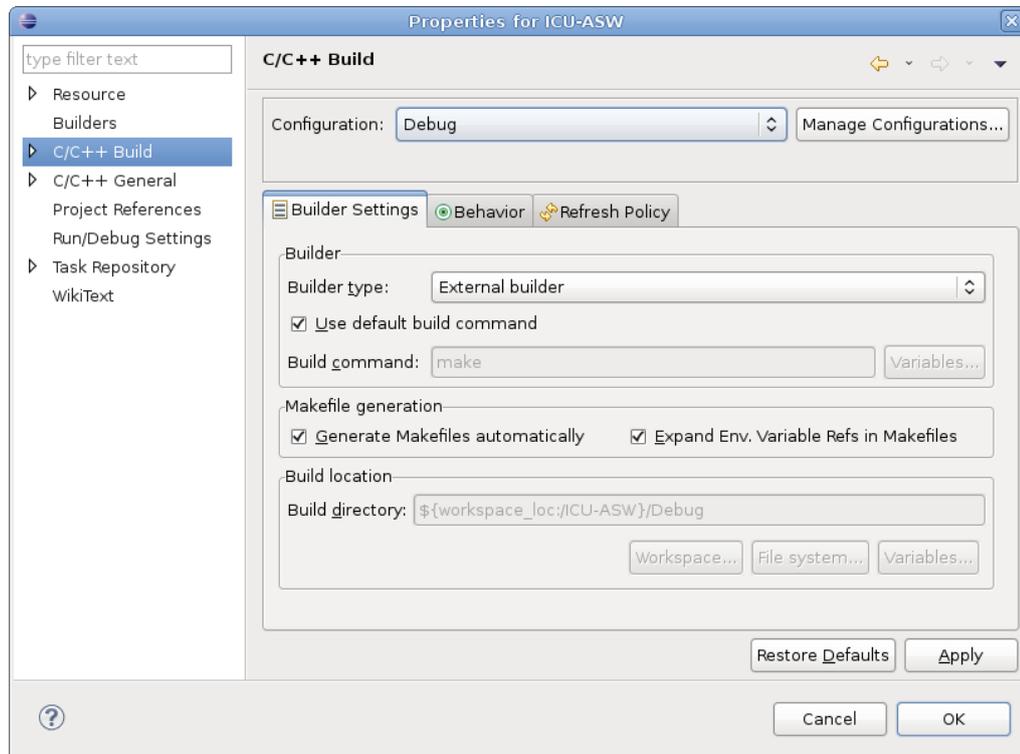


Figure B.5: *Properties window for the Eclipse ICU-ASW project.*

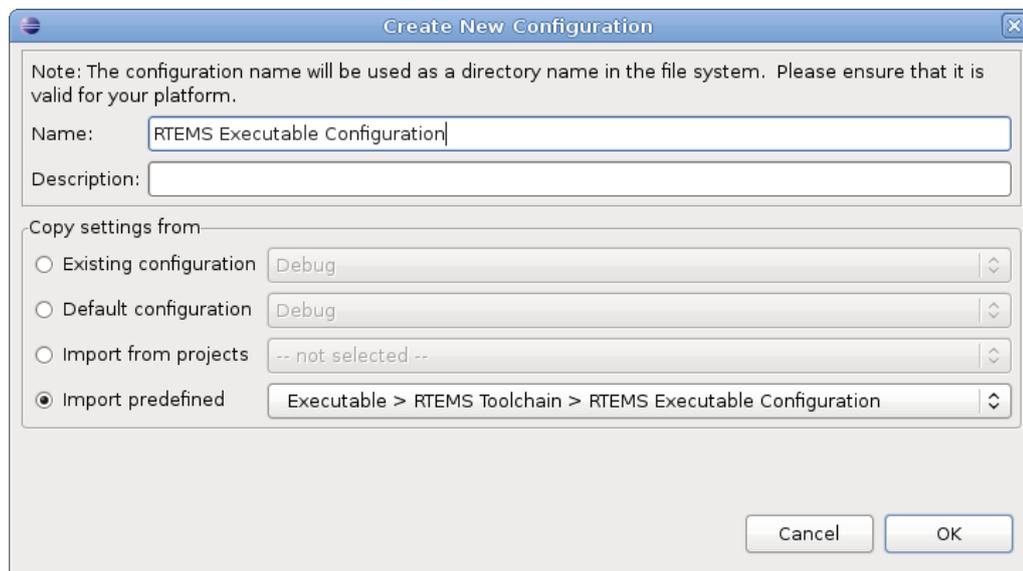


Figure B.6: *Create new configuration for the Eclipse ICU-ASW project.*

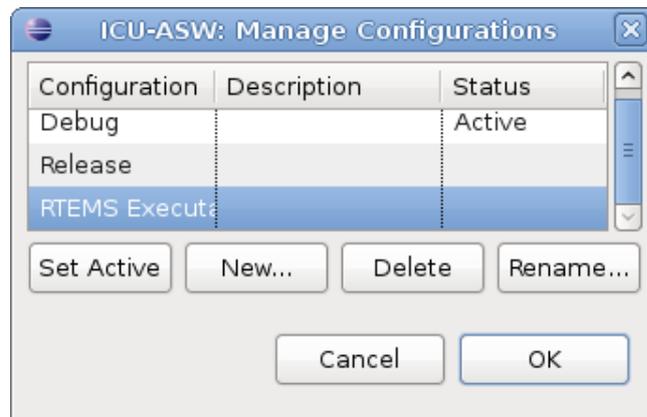


Figure B.7: Enable RTEMS configuration for the Eclipse ICU-ASW project.

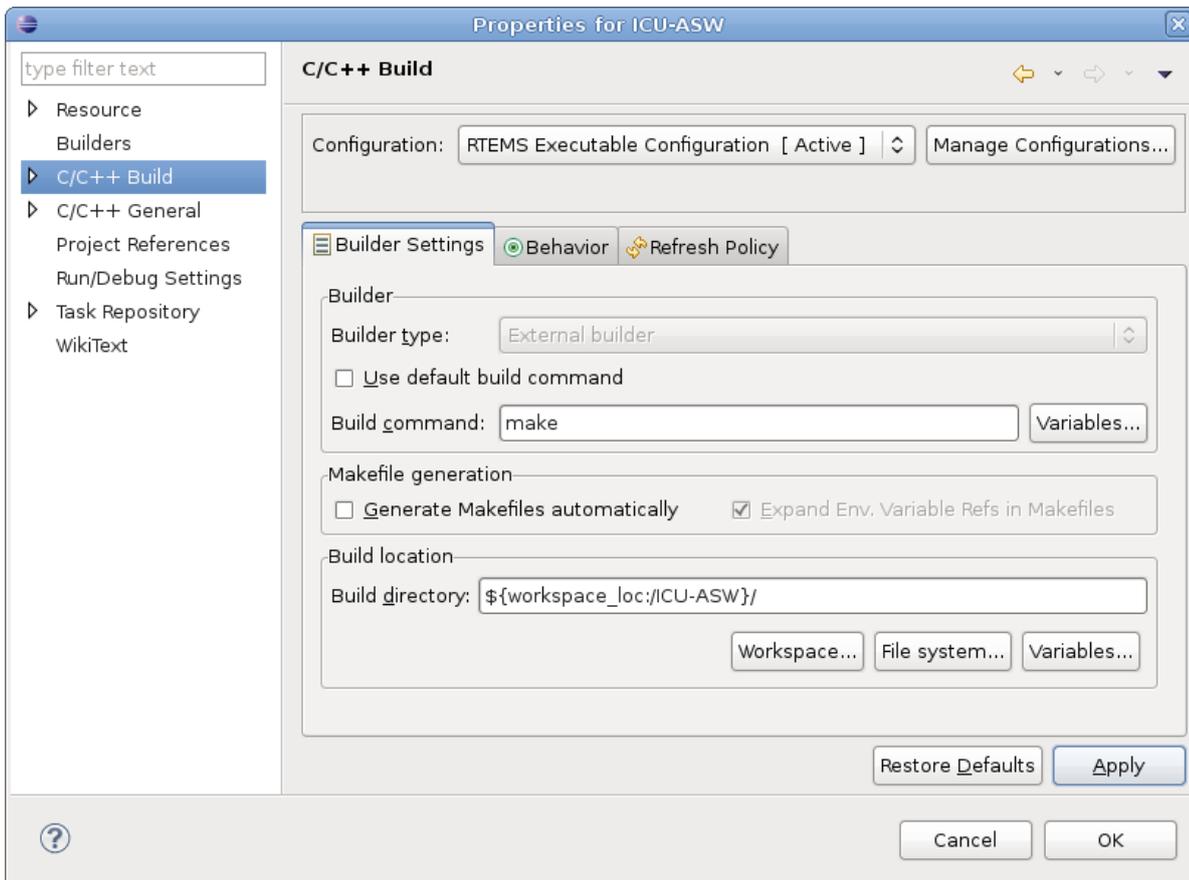


Figure B.8: Deselect *Use default build command* and *Generate Makefiles automatically* options in the Eclipse ICU-ASW project.

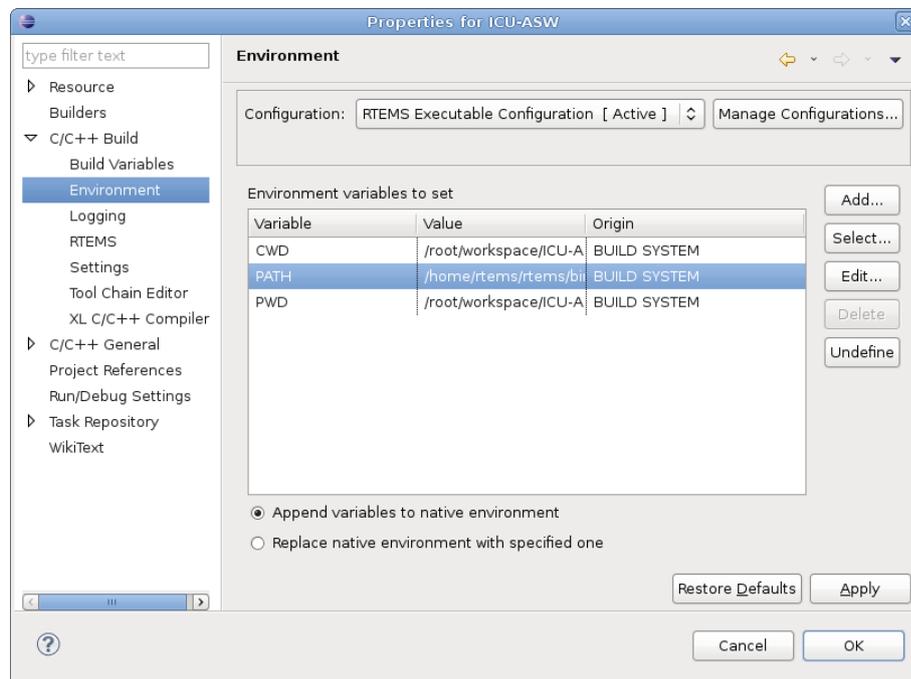


Figure B.9: Add  $\$PATH$  variable to the Eclipse ICU-ASW project environment.

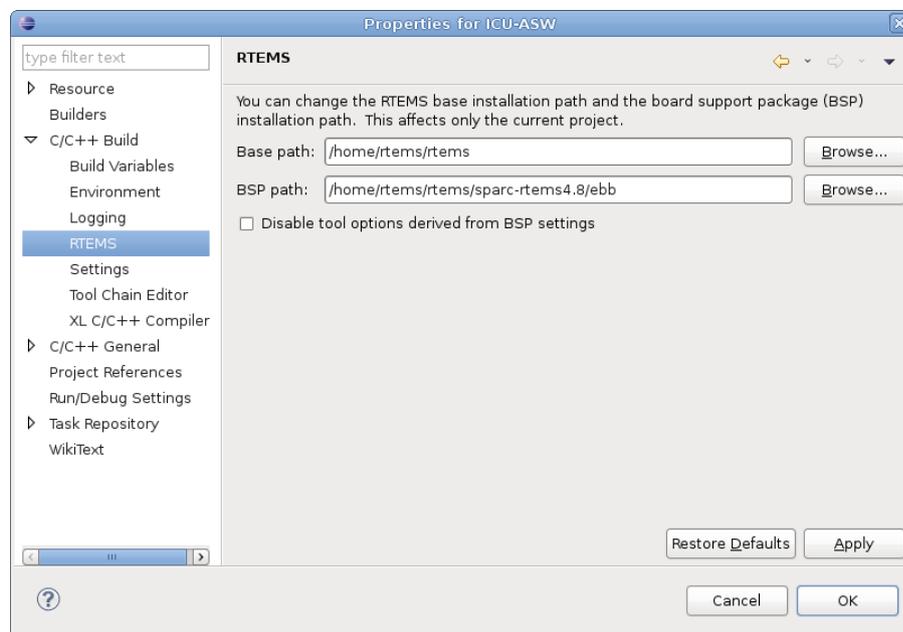


Figure B.10: Set the correct path to RTEMS and EBB Board Support Package for the Eclipse ICU-ASW project.

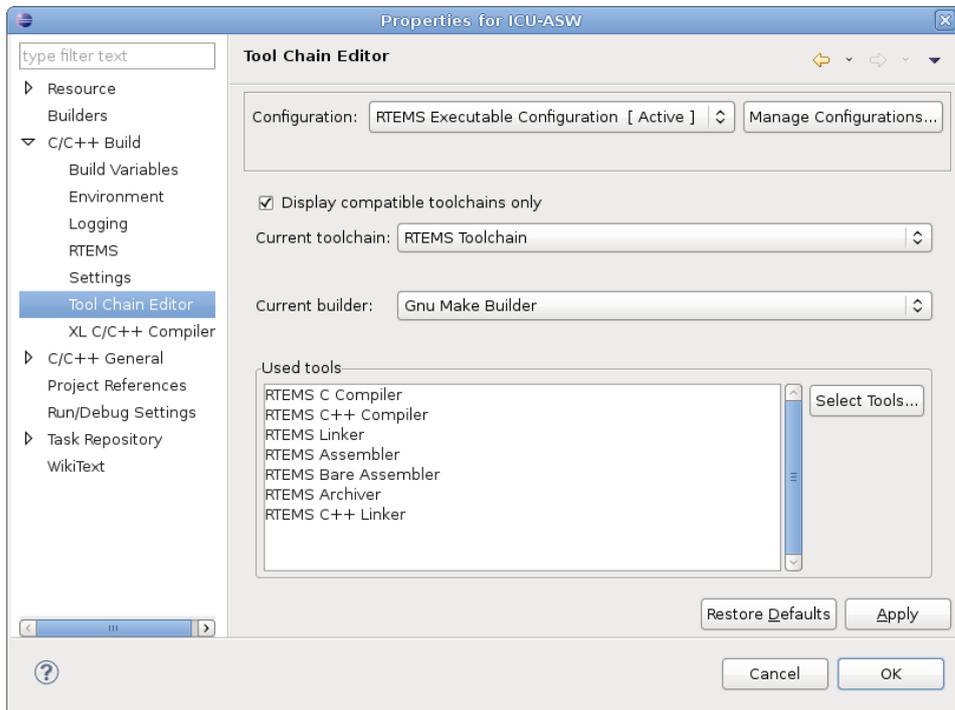


Figure B.11: Check the Toolchain setting for the Eclipse ICU-ASW project.

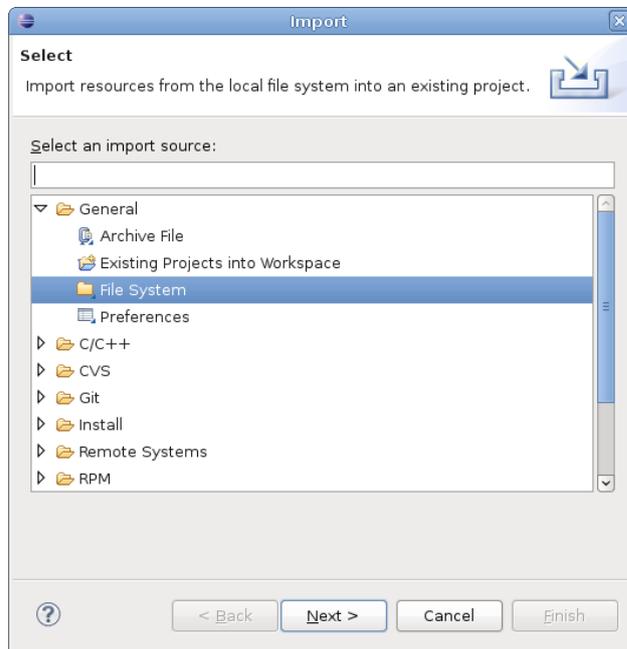


Figure B.12: Import the ICU-ASW source code in the Eclipse ICU-ASW project.

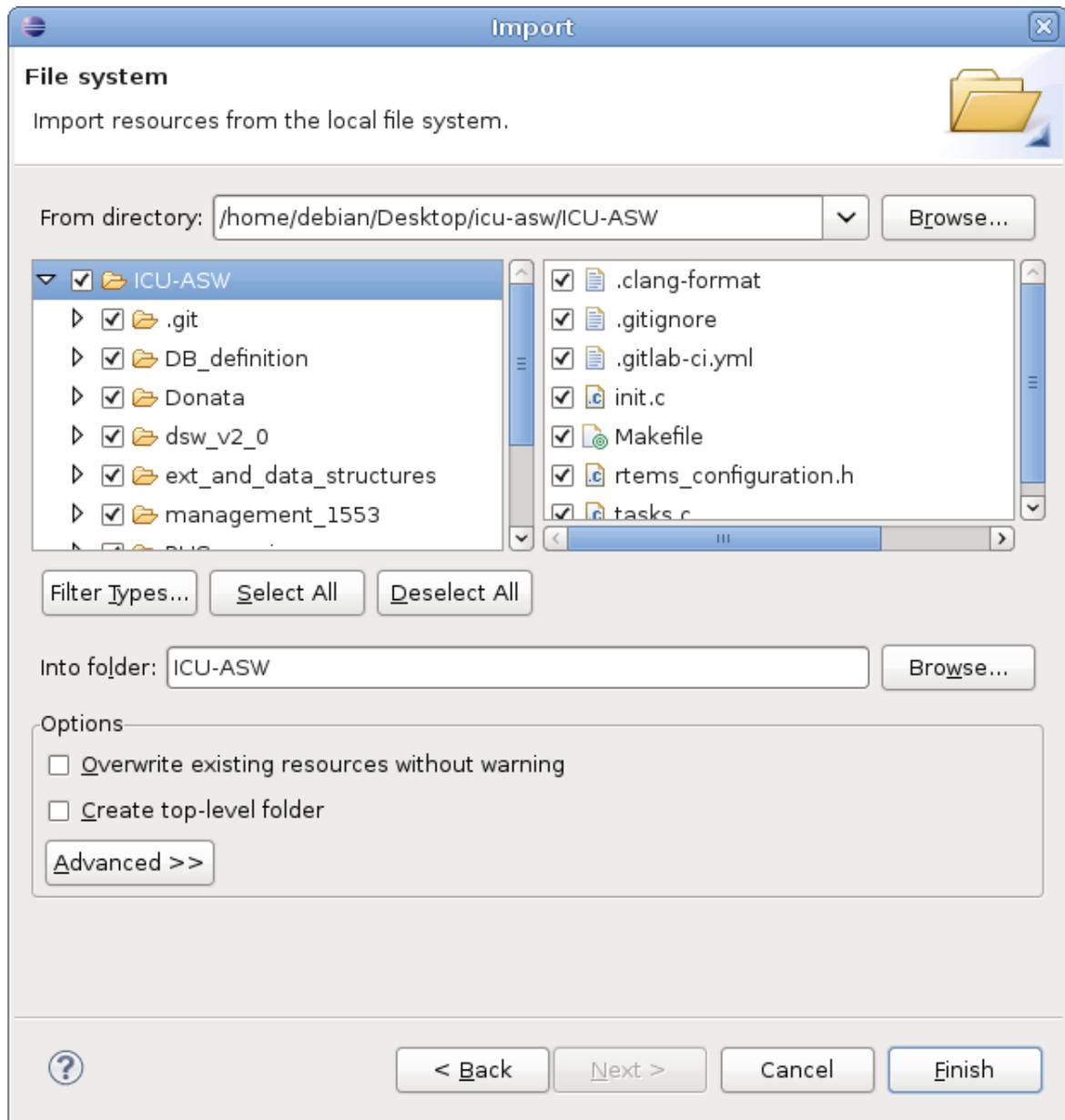


Figure B.13: Select the subdirectories to import the ICU-ASW source code in the Eclipse ICU-ASW project.

*Workbench*, on the left. Click with the right button of the mouse on the project and a drop down menu will open, then click on **Properties** and a new window will open (Figure B.5). Select **C/C++ Build** and then click on **Manage Configurations... → New...** to create a **RTEMS Executable Configuration** via the **Import predefined** option (Figure B.6). Finally, select **RTEMS Executable Configuration** on the **Manage Configurations...** menu and click on **Set Active** (Figure B.7). In the **Builder Settings** field, deselect the **Use default build command** and **Generate Makefiles automatically** options (Figure B.8). In **C/C++ Build** click on **Environment** and then on **Add...** in order to add the **\$PATH** variable (Figure B.9). In **C/C++ Build** also click on **RTEMS** and set the correct path to **RTEMS** and **EBB Board Support Package** installation folders (Figure B.10). In conclusion, check that the enabled **Toolchain** is **RTEMS Toolchain** (Figure B.11). The ICU-ASW project thus created does not include the ICU-ASW source code yet. To import the source code, just click with right mouse button on the project visible in the *Project Explorer* and select **Import**, then a window will open. Select **General → File System** and click on **Next >** (Figure B.12). The directory with the ICU-ASW source code must be inserted: click on **Browse...**, select all the subdirectories in the source code directory and click on **Finish** (Figure B.13). Last step is building the executable file. Right click on **ICU-ASW** in the *Project Explorer* and in the drop down menu just select **Clean Project** (Figure B.14), then in the same drop down menu click on **Build Project** (Figure B.15) and the executable file named **asw\_drv\_vx.xx.exe** will be created in the **o-optimize** subdirectory.

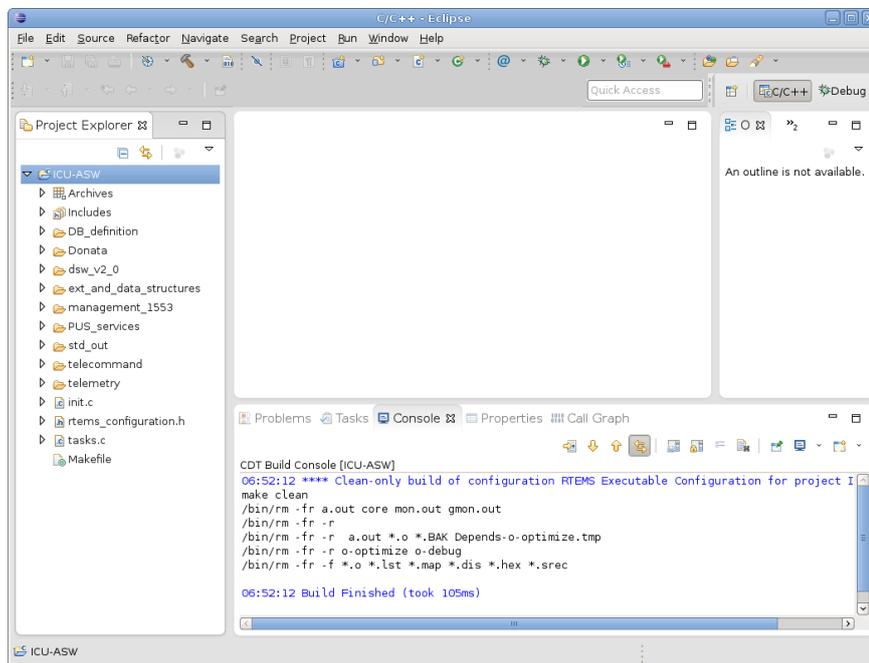
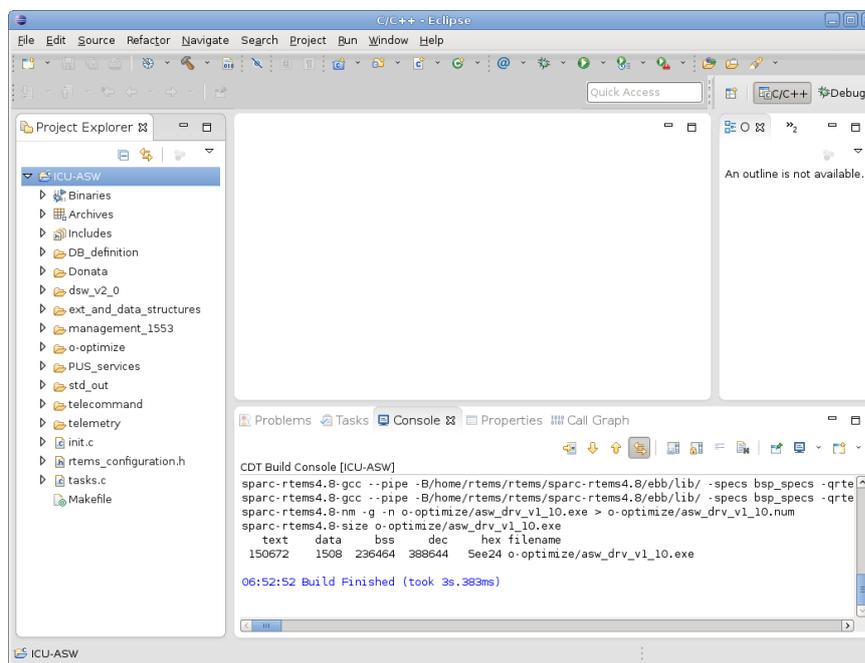


Figure B.14: Clean the Eclipse ICU-ASW project.

Figure B.15: *Build the Eclipse ICU-ASW project.*

### B.1.5 Loading and running the ICU-ASW on the *EBB* with *GRMON*

To load and run the ICU-ASW executable on the *EBB*, the *GRMON* software is needed. *GRMON* can be downloaded at <http://www.gaisler.com/products/grmon> together with its *User Manual* and the *runtime drivers (HASP HL license Drivers)* for the USB flash drive containing the *GRMON* execution licence. The *GRMON* installation procedure can be found in the 1.4 section of its *User Manual*. Once the licence drivers are installed<sup>3</sup>, the installation path of the *GRMON* executable must be added to \$PATH environment variable. The *GRMON* version currently used is 1.1.20, since it shows better compatibility with UART device. Once installed, *GRMON* can be run with the `grmon -leon2` command (the licence USB flash drive must be mounted). *GRMON* will try to communicate with the *EBB LEON2* processor, and if the *EBB* is on and the serial connection between the *EBB setup board* and the PC is installed, *GRMON* will recognize the processor and retrieve the prompt in Figure B.16. To load the ICU-ASW executable on the *LEON2* processor it is sufficient to launch the `load` command followed by the executable path (Figure B.17). Finally, to run it, the `run` command must be issued (Figure B.18). While the executable is running, *GRMON* keeps waiting. To verify that the ICU-ASW is running, it is sufficient to check the value of the current absorbed by the *EBB*: normally it is 90 mA, but when a process is running it raises up to 130 mA (Figure B.19). To

<sup>3</sup>To install the licence drivers, run the script at [https://supportportal.gemalto.com/csm/?id=kb\\_article&sys\\_id=935383e8db2ec3c8fe0aff3dbf961969](https://supportportal.gemalto.com/csm/?id=kb_article&sys_id=935383e8db2ec3c8fe0aff3dbf961969) and then the script at [https://supportportal.gemalto.com/csm/?id=kb\\_article&sys\\_id=9ecc07ecdbaec3c8fe0aff3dbf961991](https://supportportal.gemalto.com/csm/?id=kb_article&sys_id=9ecc07ecdbaec3c8fe0aff3dbf961991).

```
nisp@pceuclid2: ~
nisp@pceuclid2:~$ grmon -leon2

GRMON LEON debug monitor v1.1.20

Copyright (C) 2004,2005 Gaisler Research - all rights reserved.
For latest updates, go to http://www.gaisler.com/
Comments or bug-reports to support@gaisler.com

using port /dev/ttyS0 @ 115200 baud

initialising .....
detected frequency: 80 MHz

Component                                Vendor
LEON2 Memory Controller                  European Space Agency
LEON2 AHB Status & Failing Addr          European Space Agency
LEON2 SPARC V8 processor                  European Space Agency
LEON2 Write Protection                   European Space Agency
LEON2 Configuration register             European Space Agency
LEON2 Timer Unit                         European Space Agency
LEON2 UART                               European Space Agency
LEON2 UART                               European Space Agency
LEON2 Interrupt Ctrl                     European Space Agency
LEON2 I/O port                           European Space Agency
AHB Debug UART                           Gaisler Research
LEON2 Debug Support Unit                 Gaisler Research

Use command 'info sys' to print a detailed report of attached cores

grlib>
```

Figure B.16: *The GRMON prompt.*

```
nisp@pceuclid2: ~
nisp@pceuclid2:~$ grmon -leon2

GRMON LEON debug monitor v1.1.20

Copyright (C) 2004,2005 Gaisler Research - all rights reserved.
For latest updates, go to http://www.gaisler.com/
Comments or bug-reports to support@gaisler.com

using port /dev/ttyS0 @ 115200 baud

initialising .....
detected frequency: 80 MHz

Component                                Vendor
LEON2 Memory Controller                  European Space Agency
LEON2 AHB Status & Failing Addr          European Space Agency
LEON2 SPARC V8 processor                  European Space Agency
LEON2 Write Protection                   European Space Agency
LEON2 Configuration register             European Space Agency
LEON2 Timer Unit                         European Space Agency
LEON2 UART                               European Space Agency
LEON2 UART                               European Space Agency
LEON2 Interrupt Ctrl                     European Space Agency
LEON2 I/O port                           European Space Agency
AHB Debug UART                           Gaisler Research
LEON2 Debug Support Unit                 Gaisler Research

Use command 'info sys' to print a detailed report of attached cores

grlib> load /home/nisp/eclipse-workspace/ICU-ASW/o-optimize/asw_drv_v1_10.exe
section: .text at 0x40000000, size 150672 bytes
section: .data at 0x40024c90, size 1504 bytes
section: .jcr at 0x40025270, size 4 bytes
total size: 152180 bytes (81.3 kbit/s)
read 705 symbols
entry point: 0x40000000
grlib>
```

Figure B.17: *The GRMON load command to load an executable file on the EBB LEON2 processor.*

```
nisp@pceclid2:~$ grmon -leon2

GRMON LEON debug monitor v1.1.20

Copyright (C) 2004,2005 Gaisler Research - all rights reserved.
For latest updates, go to http://www.gaisler.com/
Comments or bug-reports to support@gaisler.com

using port /dev/ttyS0 @ 115200 baud

initialising .....
detected frequency: 80 MHz

Component                               Vendor
LEON2 Memory Controller                 European Space Agency
LEON2 AHB Status & Failing Addr         European Space Agency
LEON2 SPARC V8 processor                 European Space Agency
LEON2 Write Protection                  European Space Agency
LEON2 Configuration register            European Space Agency
LEON2 Timer Unit                       European Space Agency
LEON2 UART                              European Space Agency
LEON2 UART                              European Space Agency
LEON2 Interrupt Ctrl                   European Space Agency
LEON2 I/O port                          European Space Agency
AHB Debug UART                          Gaisler Research
LEON2 Debug Support Unit                Gaisler Research

Use command 'info sys' to print a detailed report of attached cores

gplib> load /home/nisp/eclipse-workspace/ICU-ASW/o-optimize/asw_drv_v1_10.exe
section: .text at 0x40000000, size 150672 bytes
section: .data at 0x40024c90, size 1504 bytes
section: .jcr at 0x40025270, size 4 bytes
total size: 152180 bytes (81.6 kbit/s)
read 705 symbols
entry point: 0x40000000
gplib> run
```

Figure B.18: The GRMON run command to run an executable file on the EBB LEON2 processor.

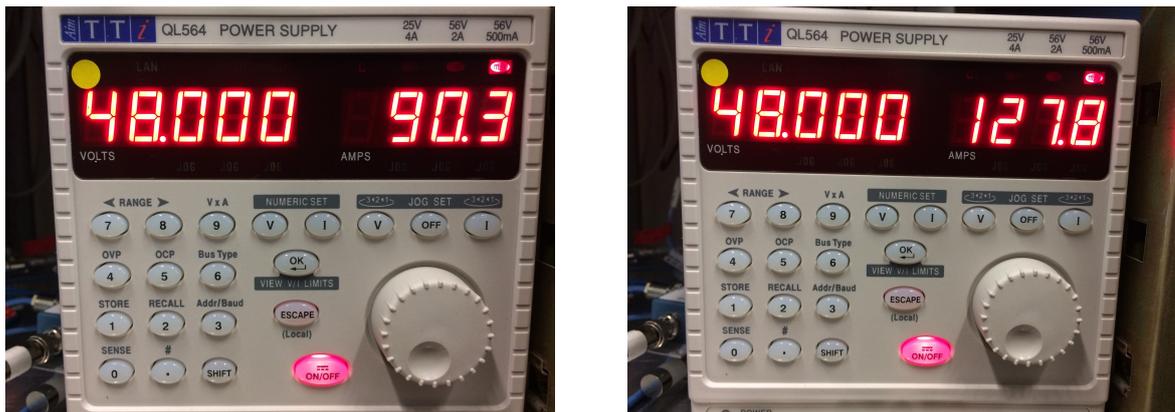


Figure B.19: On the left, the EBB powered on without running processes. On the right, the EBB powered on with ICU-ASW running.

stop the execution, the **Ctrl+C** combination must be issued and then *GRMON* will retrieve the prompt (Figure B.20). Finally, to stop *GRMON* just run the quit command (Figure B.21).

```
nisp@pceuclid2: ~
nisp@pceuclid2:~$ grmon -leon2

GRMON LEON debug monitor v1.1.20

Copyright (C) 2004,2005 Gaisler Research - all rights reserved.
For latest updates, go to http://www.gaisler.com/
Comments or bug-reports to support@gaisler.com

using port /dev/ttyS0 @ 115200 baud

initialising .....
detected frequency: 80 MHz

Component                                Vendor
LEON2 Memory Controller                   European Space Agency
LEON2 AHB Status & Failing Addr           European Space Agency
LEON2 SPARC V8 processor                   European Space Agency
LEON2 Write Protection                     European Space Agency
LEON2 Configuration register              European Space Agency
LEON2 Timer Unit                           European Space Agency
LEON2 UART                                 European Space Agency
LEON2 UART                                 European Space Agency
LEON2 Interrupt Ctrl                       European Space Agency
LEON2 I/O port                             European Space Agency
AHB Debug UART                             Gaisler Research
LEON2 Debug Support Unit                   Gaisler Research

Use command 'info sys' to print a detailed report of attached cores

grrlib> load /home/nisp/eclipse-workspace/ICU-ASW/o-optimize/asw_drv_v1_10.exe
section: .text at 0x40000000, size 150672 bytes
section: .data at 0x40024c90, size 1504 bytes
section: .jcr at 0x40025270, size 4 bytes
total size: 152180 bytes (81.3 kbit/s)
read 705 symbols
entry point: 0x40000000
grrlib> run
^C
Interrupt!
stopped at 0x4000186c
grrlib>
```

Figure B.20: The *GRMON stop* command to stop the execution of a process running on the EBB.

### B.1.6 Running the ICU-ASW in *Debug mode* with *Eclipse*

To run the ICU-ASW in *Debug mode* with the *LEON Debugger* in *Eclipse*, the installation of *GDB 6.6* with *Edisoft patch* is required.

#### Building *GDB 6.6* with *Edisoft patch*

*GDB 6.6* must be built on 32-bit *Debian 5* virtual machine with *RTEMS* installed. The *GDB* source code and the *Edisoft patch* are on the CD provided by *Edisoft*. Once the source code and the patch are copied on the virtual machine, *GDB* can be built as follows:

- open a terminal window and cd to `/home/rtems/build`;

```

nisp@pceuid2:~
nisp@pceuid2:~$ grmon -leon2

GRMON LEON debug monitor v1.1.20

Copyright (C) 2004,2005 Gaisler Research - all rights reserved.
For latest updates, go to http://www.gaisler.com/
Comments or bug-reports to support@gaisler.com

using port /dev/ttyS0 @ 115200 baud

initialising .....
detected frequency: 80 MHz

Component                               Vendor
LEON2 Memory Controller                 European Space Agency
LEON2 AHB Status & Failing Addr         European Space Agency
LEON2 SPARC V8 processor                 European Space Agency
LEON2 Write Protection                   European Space Agency
LEON2 Configuration register            European Space Agency
LEON2 Timer Unit                        European Space Agency
LEON2 UART                               European Space Agency
LEON2 UART                               European Space Agency
LEON2 Interrupt Ctrl                    European Space Agency
LEON2 I/O port                           European Space Agency
AHB Debug UART                           Gaisler Research
LEON2 Debug Support Unit                 Gaisler Research

Use command 'info sys' to print a detailed report of attached cores

grib> load /home/nisp/eclipse-workspace/ICU-ASW/o-optimize/asw_drv_v1_10.exe
section: .text at 0x40000000, size 150672 bytes
section: .data at 0x40024c90, size 1504 bytes
section: .jcr at 0x40025270, size 4 bytes
total size: 152180 bytes (81.6 kbit/s)
read 705 symbols
entry point: 0x40000000
grib> run
^C
Interrupt!
stopped at 0x40002808
grib> quit
nisp@pceuid2:~$

```

Figure B.21: The GRMON quit command to end a GRMON session.

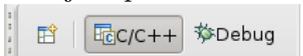
- here run the uncompression command `tar xjvf /path/to/gdb-6.6.tar.bz2;`
- to apply the *Edisoft* patch run  
`patch --dry-run -p0 < /path/to/gdb-6.6-rtems4.8-20071216.diff`  
and then, if no errors are generated,  
`patch -p0 < /path/to/gdb-6.6-rtems4.8-20071216.diff;`
- cd to `/home/rtems/build/gdb-6.6` and run `./configure --target=sparc-rtems4.8;`
- once the configuration is completed, run `make all` and wait for the end of the execution;
- the `gdb-6.6` directory should be ready and can be transferred to any machine with *RTEMS* installed (it has to be copied to `/home/rtems/build`).

In the case of the current Ubuntu 16.04 host machine, the `gdb-6.6` directory has to be copied to `/home/rtems/build` and then the installation path of *GDB* (`/home/rtems/build/gdb-6.6/gdb`) must be added to `$PATH` environment variable.

### Starting the ICU-ASW in *Debug mode*

To start the ICU-ASW in *Debug mode* with *Eclipse*, right click on the ICU-ASW project in the *Project Explorer* and then click on *Debug As* → *Debug Configurations*. Select *LEON C/C++ application* and name the debug configuration as *ICU-ASW Debug*; check that the *Build*

configuration is set to RTEMS Executable Configuration (Figure B.22). At this point, click on Debugger and insert the word `Init` in `Stop on startup at:.` In Debugger Options set the *GDB* installation path to `/home/rtems/build/gdb-6.6/gdb/gdb`, IP Address to the IP of the PC running *GRMON* and Port Number to 2222 (Figure B.23). *GRMON* has to be run with the command `grmon -leon2 -gdb`, so that it will start a server waiting for *GDB* on port 2222. Clicking on Debug in the Debug Configurations window, *Eclipse* will communicate with *GRMON* via *GDB* and start the ICU-ASW application in *Debug mode*. The window in Figure B.25 will open and the execution will stop at the beginning of the `Init` routine. The execution will be resumed by clicking on `F8`. Alternatively the application can be executed step by step with *Step Over* (`F6`), *Step Into* (`F5`) and *Step Return* (`F7`). Breakpoints can be inserted at a given line with `Shift+Ctrl+B`. To stop the execution at any time just press `Ctrl+F2`. To

get back to the software development window just press on the `→`  icon on top right.

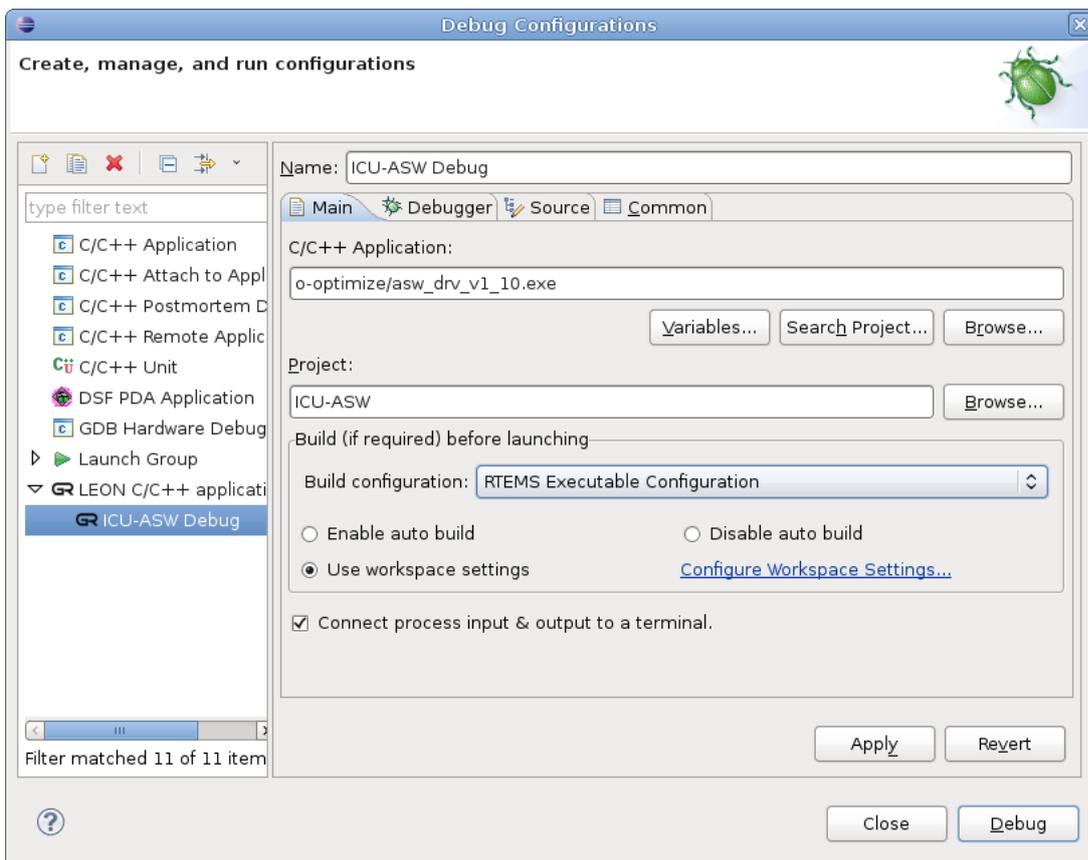


Figure B.22: Set Debug configurations for the Eclipse ICU-ASW project.

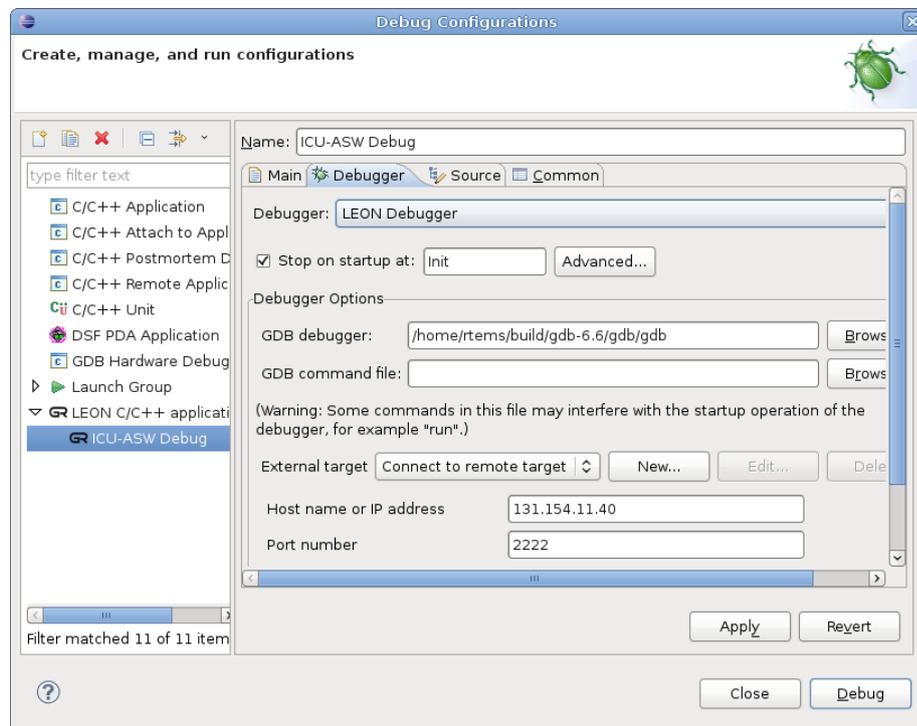


Figure B.23: Set Debugger configurations for the Eclipse ICU-ASW project.

```
nisp@pceuclid2: ~
nisp@pceuclid2:~$ grmon -leon2 -gdb

GRMON LEON debug monitor v1.1.20

Copyright (C) 2004,2005 Gaisler Research - all rights reserved.
For latest updates, go to http://www.gaisler.com/
Comments or bug-reports to support@gaisler.com

using port /dev/ttyS0 @ 115200 baud

initialising .....
detected frequency: 80 MHz

Component                               Vendor
LEON2 Memory Controller                 European Space Agency
LEON2 AHB Status & Failing Addr         European Space Agency
LEON2 SPARC V8 processor                 European Space Agency
LEON2 Write Protection                  European Space Agency
LEON2 Configuration register            European Space Agency
LEON2 Timer Unit                        European Space Agency
LEON2 UART                              European Space Agency
LEON2 UART                              European Space Agency
LEON2 Interrupt Ctrl                   European Space Agency
LEON2 I/O port                          European Space Agency
AHB Debug UART                          Gaisler Research
LEON2 Debug Support Unit                Gaisler Research

Use command 'info sys' to print a detailed report of attached cores
gdb interface: using port 2222
```

Figure B.24: GRMON running with `-gdb` option to enable a GDB server on port 2222.

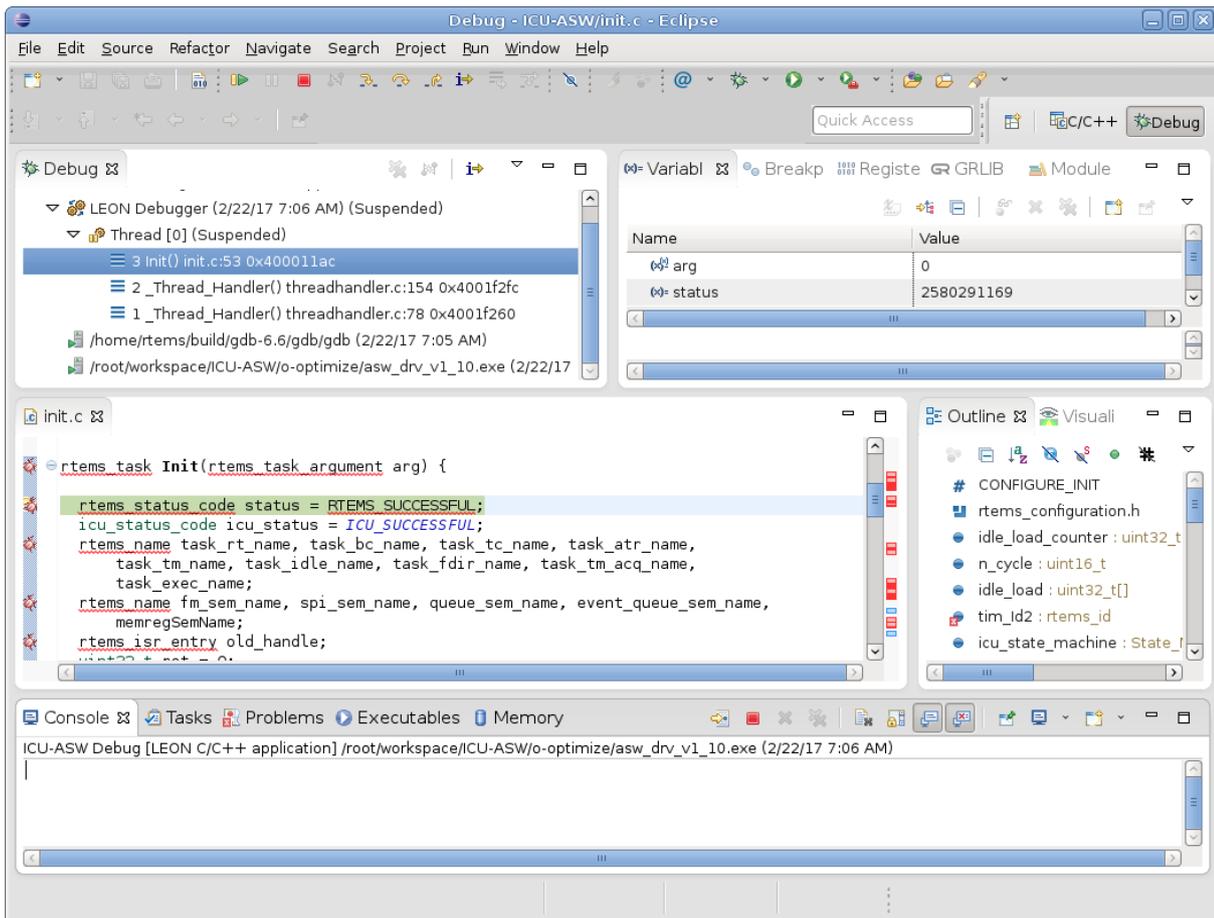


Figure B.25: The Eclipse Debug interface for the ICU-ASW project.

## B.2 Building and running *Edisoft RTEMS* tests

*Edisoft RTEMS* tests source code is at the *Git* repository <https://baltig.infn.it/euclid/rtems-testsuite.git>. The *Git* project comprises 3 branches:

- *master*, containing the original *Edisoft testsuite*;
- *ebb*, including in the *testsuite* the modifications to make the source code compatible with the *EBB Board Support Package* and the *GRMON* communication interface;
- *quiet*, with a modification in the source code to make `printk()` routine write only the test final reports, in order to reduce the *testsuite* execution time.

The *testsuite* must be executed on a 32-bit *Debian 5* virtual machine with *RTEMS* installed.

### B.2.1 Building and running the *testsuite*

To run the *testsuite* on the *Debian 5* virtual machine, *GRMON* must be installed also on the virtual machine itself. Version 1.1.20 was used because of its compatibility with UART device. In order to run *GRMON* on the virtual machine, the *runtime drivers* for the licence USB flash drive must be installed. To mount the USB flash drive on the virtual machine, click on **Devices** → **USB** → **AKS HASP HL 3.25 [0325]** on *VirtualBox* menu bar. Once the *testsuite* folder is downloaded from the *Git* repository, it has to be copied to `/home/rtems/build/rtems-impr`. Then open a terminal window, `cd` to

```
/home/rtems/build/rtems-impr/testsuites/support/scripts
```

and build and run the entire *testsuite* launching the command

```
./run_test_suite.pl --deploydir=/path/to/deploydir --edilibpath=/opt/edilib  
--targetplat=TargetBoard1EBB --build. In deploydir the executable files will be placed,  
then the Perl script will communicate with GRMON in order to load and run the executables  
on the EBB LEON2 processor. After the execution of each test, a report will be generated  
and placed in a deploydir subdirectory. Edisoft tests are divided into 4 sections: host, stan-  
dard, coverage and performance. In testsuites/support/scripts folder, the Perl scripts  
run_test_suite.pl, run_host.pl, run_std.pl, run_cov.pl and run_perf.pl are present.  
Each script builds and runs only the tests belonging to the corresponding section.
```