THE MIRI IMAGER GROUND SUPPORT EQUIPMENT CONTROL SYSTEM BASED ON PCS

Francoise Gougnaud, Denis Arranger, Philippe De Antoni, Gilles Andre Durand, Dominique Eppelle, Alain Goetschy, Yves Lussignol, Pierre Mattei, CEA Saclay Dapnia, Gif sur Yvette, France

Abstract

The James Web Space Telescope (JWST) is the successor of Hubble in the infrared. Our division, Dapnia, is in charge of the design and completion of the optomechanical part of the imager called MIRIM, one instrument of JWST, and of its test bench called the Ground Support Equipment (GSE). This GSE consists of a warm telescope simulator, of a model (identical to the flight model) of the imager, of a cryostat to cool the imager down to its operating temperature, and of an infrared detector (1024x1024 pixels). The telescope simulator is composed of several optical components to control (hexapod, mirrors, etc.). The major part of the hardware architecture for the control of the IR detector and the telescope simulator is based on PCs and COTS boards. This paper describes the software development and its specificities. ESO software (IRACE and BOB) and EPICS are associated to complete the operator interface. The cryostat control is our homemade supervision system for cryogenics systems based on PLCs, on the WorldFIP Fieldbus network, and on an industrial XPe PC. The tests of the different subsystems have started, and the whole test bench will be operational in summer 2007.

THE MIRI IMAGER GSE OVERVIEW ARCHITECTURE CONTROL

The first goal of this test bench is to perform cold tests (at 7°K) of the imager MIRIM. It comprises several subsystems:

- Filter Wheel Control
- Warm Telescope Simulator Control
- Infrared Detector Slow Control
- Infrared Detector Acquisition
- Cryostat Control
- One PC dedicated to the high level operator interface and operation sequencing.

All these subsystems run on PCs except the slow control of the infrared detector which runs on VME.

Three software cohabit: the ESO software IRACE (Infrared Array Control Electronics) [1], EPICS and the homemade Anibus-FBI PLCs supervision [2].

Here is an overview of these different software. IRACE controls the infrared detector acquisition, EPICS controls the Filter Wheel, the Warm Telescope Simulator and the Slow Control of the infrared detector. The Anibus-FBI software controls the cryostat. The different subsystems communicate thanks to Channel Access. BOB (Broker for Observation Blocks) is the high level user interface software providing the supervisory control. It allows to describe and to execute the different steps of an experiment and to carry out successive exposures.

FILTER WHEEL CONTROL

The Filter Wheel has 18 positions with different optical elements (filters, imaging filters and prism). Max Planck Institute of Astronomy (MPIA) from Heidelberg was in charge of the control for this subsystem. They develop their own control, based on VISUAL C++. It consists in moving the wheel stepwise through the positions, read the position and temperature sensors. Consequently for Dapnia, the problematic was to build an interface between the MPIA Wheel control and EPICS.

The MPIA Host PC is a Pentium 4.3 GHz Windows XP. The I/O board used to control the motor is a Meilhaus ME-4680, 32 A/D channels, 4 D/A channels, 32 digital I/O. The Meilhaus board, in fact, controls an electronics box that essentially comprises:

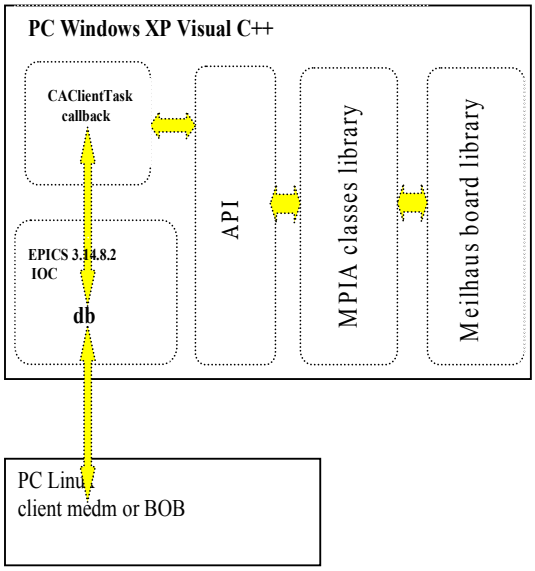
- a Linear Amplifier Board which generates a controlled output current for driving the C84 motor and measures the motor's current and voltage
- a Position Sensor Board which reads the position and the temperature sensors.

MPIA has provided us with an API to control the hardware. This API functions can initialize the equipment, get the position, get the temperature, move the wheel, get the movement status.

Therefore, we have built a Windows EPICS soft IOC (3.14.8.2) based on VISUAL C++ .NET 7.0. All the steps are described in the document [5]. This IOC manages a database. The starting Process Variable is the asked wheel position between 1 and 18. This record is processed whenever a client medm or BOB changes its value.

A client channel access task has been developed. This client channel access task subscribes a state change with ca_create_subscription and specifies the callback to be invoked whenever the position variable value

changes. This callback calls the API functions to run a movement, read the position, give different status, … and refreshes the different process variables (movement status, error numbers, error messages, position, temperature, …).



WARM TELESCOPE SIMULATOR CONTROL

It needs to control different devices: a HGH extended infrared light source, a Lot-Oriel Cornerstone motorized monochromator, a Symetrie positioning hexapod [3] for alignment and a Newport Motion Controller XPSC8.

The goal of the extended infrared source is to get the temperature regulation of a blackbody in absolute or differential mode.

The Cornerstone monochromator 74000 packs two gratings and an integrated shutter.

The hexapod was specially designed for our telescope simulator. It can move in 6 degrees of freedom, in 3 translations and 3 rotations with high accuracy: resolution of 1µm in translation and 4 µrad in rotation. Its controller is an industrial PC with XP Embedded and Labview7 and equipped with a GALIL board to drive the motor.

The Newport XPSC8 Motion Controller that drives mirrors and four motors for the pupil is itself a Pentium 4 PC based, has an Ethernet control interface and a digital and analog I/O built-in. The EPICS Motor Record has been used.

A PC control subsystem is dedicated to this telescope. It is equipped with a QUATECH PCI board that provides 4 serial lines. This PC can be stand alone and communicate with the main control. A medm screen is dedicated to each device. It runs EPICS (R3.14.8.2) under Linux (RHEL4) on a host-based IOC and the controls for each device are carried out with the streamDevice package(2-1) based on the asyn driver (4-6). Protocol files include the different commands taking advantage of the various possibilities of streamDevice:

- a checksum is needed by infrared extended source, it's accomplished using the bitwise inverse of the sum and changing representation from binary to hexadecimal ascii
  ADD = 0x81; DC2 = 0x12;
  TsetPoint { out $ADD,$DC2,"%+05d%0<-sum>";}
- relating to the hexapod, the six motions (x, y, z, Rx, Ry , Rz) have to be done simultaneously in the same string statement: this is done with a waveform (input record) which outputs its data.
  record(waveform, wfout) {
  field(DTYP, "stream")
  field(INP, "@hexapod.proto wave_hexapod COM6")
  field(NELM, "6")
  field(FTVL, "STRING")
  }
  wave_hexapod {Separator = ","; out "MOV%s";}
  The array is filled up and sent to the device in a SNL program.

Therefore, only configuration was needed to control the different devices for this warm telescope simulator and EPICS has permitted to provide quickly this subsystem control.

INFRARED DETECTOR CONTROL

Acquisition

The software package IRACE from ESO, that the lab had already used for the VISIR instrument (VLT Imager and Spectrometer for mid InfraRed) [3] was chosen and has been ported on the selected COTS boards. Originally, the controller IRACE consists of a software package and electronic boards designed by ESO. For the MIRI test bench, SPECTRUM boards were selected to replace the ESO's boards. They run in an industrial PC (RHEL4).
Two SPECTRUM MI7011 boards are needed. The first is dedicated to the readout of the detector. The second generates the signal to start the digitized acquisition of the infrared detector frame. The SPECTRUM MI4541 board performs the acquisition itself. The acquisition hardware and software architecture is described in [4] and [7].

EPICS Channel Access has been installed on this industrial Linux PC and allows the communication with

the detector biases and clocks levels slow control subsystem.

Slow Control

The setting for the thirteen biases and clocks levels voltages has been carried out on an EPICS VME IOC. The milestones at the beginning of the project lead us to choose a solution already tested in our Lab based on the VME ADAS boards.

To set up biases and clocks, a front-end electronic board has been designed with 13 daisy-chained DACs to limit CEM problems. Data are written by a SNL program through 2 SPI busses, each SPI bus being composed of 3 channels of the binary input/output ICV296 card. The SPI bus minimizes the number of links. Biases and clocks are permanently monitored by the SNL that entirely supports the safety of the detector. The SNL program monitors also a possible alarm of the detector temperature provided by the electronic front-end and in this case stops biases and clocks.

BOB SUPERVISION

The ESO software tool BOB (Broker for Observation Blocks) is the high-level interface towards the Detector Control and Acquisition System and the different subsystems Warm Telescope Simulator and Filter Wheel. It operates on Observation Blocks Descriptions (OBDs), read from disk files, containing all the information needed to execute exposures. BOB's task is to break OBDs down to individual exposures, sending sequences of commands to MIRIM Software.

Experimental procedures for MIRIM are written as Tcl scripts named Templates in ESO terminology. OBDs describe what Templates to call with which parameters value. Several Tcl object classes have been written to access easily to the different systems and devices from Templates. These classes include a simulation mode allowing to test the Templates without needing the availability of any hardware.

CRYOSTAT CONTROL

It relies on the “Anibus-FBI” supervision developed in our Lab for the Dapnia cryogenics installations. The architecture is based on real-time fieldbus WorldFIP, PLCs and an industrial XPe PC Web server. It was detailed in [2]. This subsystem successfully started several months ago. A last step to do is the communication between this control with the detector subsystem that gets detector temperatures read out by a Lakeshore 340 temperature controller.

CONCLUSION

The hardware architecture is based on PCs and the choice of using large community software like EPICS and BOB/IRACE for this test bench was very convenient. The communication between the different subsystems with Channel Access was easily performed. The scientific tests will take place at the beginning of 2008.

ACKNOWLEDGEMENTS

The author wishes to acknowledge the contributions of Philippe Beauvais and Joachim Neves-Da-Costa for the design of the racks, Vincent Moreau and Samuel Ronayette for their advices for the users interfaces.

REFERENCES

[1] IRACE the ESO Controller for Infrared Arrays www.eso.org/projects/iridt/irace

[2] C. Walter, “supervision Web avec Java, c’est possible” Jautomatise septembre 2004

[3] JF Gournay and Y. Lussignol “Configuration and database for the Control System of the VISIR instrument” Icalepcs01 San Jose, 27-30 November 2001

[4] F. Gougnaud, Y. Lussignol & al “The control system for the MIRI imager Ground Support Equipment” Icalepcs 2005

[5] F. Gougnaud , MIRIM Ground Support Equipment Control System Specifications MIRI-SP-00444-CEA

[6] French company in southern France http://www.symetrie.fr

[7] Y. Lussignol, The Infrared Detector Control System MIRI-DD-00156-CEA

[8] Dirk Zimoch, “Stream Device 2” http://epics.web.psi.ch/software/streamdevice/