CARMENES: The CARMENES instrument control software suite. I

J. Colomé*[a], J. Guàrdia[a], H.-J. Hagen[b], R. Morales Muñoz[c], M. Abril[c], D. Benítez[h], J. A. Caballero[d,e], M. L. Fresno[e], A. García-Piquer[a], Ll. Gesa[a], E. de Guindos[h], E. de Juan[h], J. Schiller[g], I. Vico[h], F. Vilardell[a], M. Zechmeister[f], A. Reiners[f], I. Ribas[a], W. Seifert[d], A. Quirrenbach[d], P. J. Amado[c]

[a]Institut de Ciències de l'Espai (IEEC-CSIC), Can Magrans, S/N, Campus UAB, E-08193, Barcelona, Spain;
[b]Hamburger Sternwarte, Gojenbergsweg 112, D-21029 Hamburg, Germany;
[c]Instituto de Astrofísica de Andalucía (CSIC), Glorieta de la Astronomía s/n, E-18008 Granada, Spain;
[d]Landessternwarte (ZAH), Königstuhl 12, D-69117 Heidelberg, Germany;
[e]Centro de Astrobiología (CSIC-INTA), Campus ESAC, Camino Bajo del Castillo s/n, E-28692 Villanueva de la Cañada, Madrid, Spain;
[f]Institut für Astrophysik (GAUG), Friedrich-Hund-Platz 1, D-37077 Göttingen, Germany;
[g]Thüringer Landessternwarte Tautenburg, Sternwarte 5, D-07778 Tautenburg, Germany;
[h]Centro Astronómico Hispano-Alemán (MPG-CSIC), Observatorio Astronómico de Calar Alto, Sierra de los Filabres, E-04550 Gérgal, Almería, Spain;

ABSTRACT

The main goal of the CARMENES instrument is to perform high-accuracy measurements of stellar radial velocities (1 m/s) with long-term stability. CARMENES is installed at the 3.5 m telescope in the Calar Alto Observatory (Spain) and it is equipped with two spectrographs covering from the visible to the near-infrared. We present the software packages that are included in the instrument control layer. The coordination and management of CARMENES is handled by the Instrument Control System (ICS), which is responsible for carrying out the operations of the different subsystems providing a tool to operate the instrument in an integrated manner from low to high user interaction level. The ICS interacts with the following subsystems: the near-infrared (NIR) and visible channels, composed by the detectors and exposure meters; the calibration units; the environment sensors; the front-end electronics; the acquisition and guiding module; the interfaces with telescope and dome; and, finally, the software subsystems for operational scheduling of tasks, data processing, and data archiving. The software control framework and all the software modules and layers for the different subsystems contribute to maximize the scientific return of the instrument. The CARMENES workflow covers from the translation of the survey strategy into a detailed schedule to the data processing routines that extract radial velocity data from the observed targets. The control suite is integrated in the instrument since the end of 2015.

Keywords: Site operations, data networks, instrument control systems, scheduler, spectrograph, pipelines

1. INTRODUCTION

CARMENES (Calar Alto high-Resolution search for M dwarfs with Exo-earths with Near-infrared and optical Echelle Spectrographs) performs high-precision measurements of stellar radial velocities with long-term stability[1, 2]. To carry out its purpose, CARMENES is based on two spectroscopic channels, optimized in the near-infrared (NIR) and visible (VIS) windows, and multiple subsystems that have to work in a coordinated manner.

The CARMENES instrument is formed by a set of hardware and software components in both instrument channels: VIS and NIR. In addition, the nominal operation of the instrument requires a fully synchronized control of the telescope and

---

* colome@ieec.cat; carmenes.caha.es

dome structure. An auxiliary set of instrumentation for environment monitoring (weather station, cloud sensor, seeing monitor, etc.) is also necessary for a suitable and efficient operation. The system is completed by the computer resources and network.

The global control system of the CARMENES instrument is organized in individual functional units called subsystems. Subsystems can just have a software entity (i.e., data pipeline) or can involve the operation of hardware. The subsystems in the CARMENES instrument are described in the following sections. Figure 1 depicts the scheme of logical and physical connections between all main CARMENES subsystems. In terms of connectivity and control, the ICS accesses the monitored data of all subsystems and provided by computers controlling NIR and VIS channels, NIR and VIS CUs, FE, A&G, Interlocks, and the CArmenes Scheduling Tool (CAST).

The ultimate control of the whole instrument is done from the Instrument Control System (ICS), including the Graphic User Interface (GUI). Each of the two channels has a main computer with redundancy that communicates with ICS through Ethernet, called NIR and VIS computers. These provide ICS with the main data and status from the different subsystems of each channel.

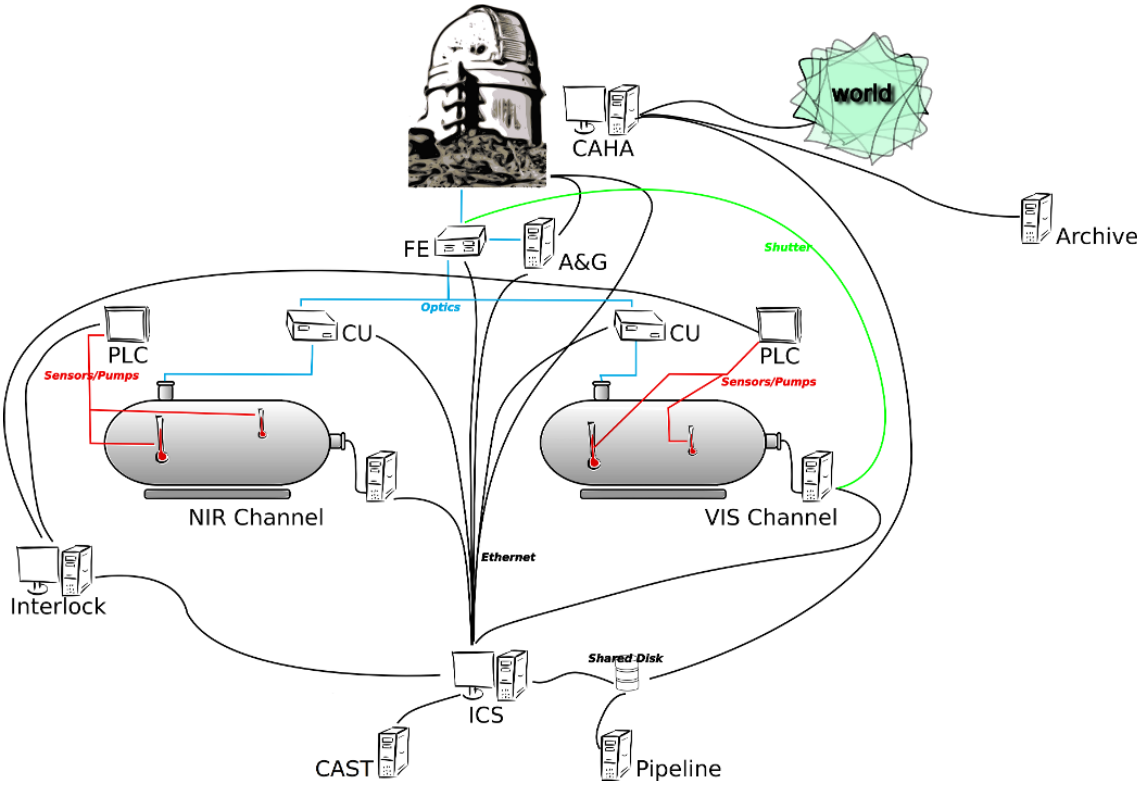


Figure 1: CARMENES System architecture. CU, A&G and FE stand for calibration unit (one for each channel), acquisition and guiding

1.1 Operation data flow

The CARMENES instrument control software suite is devoted to guarantee the operational design and the survey observational strategy[3]. This is finally translated in a specific closed-loop data flow that goes from *science* to *science* via *engineering*. In short, astronomers set the input (top-level requirements and target list) and output (scientific results; in our case exoplanet orbital parameters and basic stellar parameters), but are not usually involved in every intermediate step. Such steps deal mostly with electronics and, especially, software. Here we briefly overview the most important steps of the CARMENES data flow, which are described in detail by Caballero et al. (2016, SPIE, this volume).

1. *Input catalogue*. Carmencita, the CARMEN[ES] Cool dwarf Information and daTa Archive, is the CARMENES input catalogue[4, 5, 6]. Built by astronomers, it contains approximately 2200 nearby M dwarfs of which we have selected the 300 best targets for radial-velocity monitoring during guaranteed time observations. The 300 target stars are the brightest M dwarfs ($J < 7.0-11.5$ mag, depending on spectral subtype) observable

from Calar Alto ($\delta > -23$ deg) without known companions at close angular separations ($\rho < 5$ arcsec). For constructing Carmencita, we have compiled astronomical data from numerous previous works, public all-sky catalogues and from our own observations (low- and high-resolution imaging and spectroscopy in the optical). While the main output of this step is the target list of 300 M dwarfs, the quantity and quality of the compiled data have allowed us to open a new window of spin-off science cases with non-CARMENES data: from stellar activity, through kinematics, to multiplicity and metallicity.

2. *Scheduler, Instrument Control System and Graphical User Interface*. The target list of 300 M dwarfs from Carmencita feeds the scheduler software. The ICS GUI provides the observer with the best M dwarf to be monitored next as computed by the scheduler with the information provided by the ICS. The GUI is designed to be as simple and friendly as possible, for easy use by any observer.
3. *Raw data and FITS headers*. The output of an observing block (an "exposure") is a set of five files per target and epoch of observation: acquisition image (*-acg.fits), VIS spectrum (*-vis.fits), VIS exposure-meter count rate (*-vis.expm), NIR spectra (*-nir.fits) and NIR exposure-meter count rate (*-nir.expm). All output files share a common string such as car-yymmddThh:mm:ss-xxx-yyyy-*. It indicates the instrument (CARMENES), year, month and day, hour, minute and second of start of observing block in Universal Time, image type –sci: science, cal: calibration, tst: test–, and four-character programme code as defined by the observatory (in open time: three first characters of surname and first character of name, e.g. colj for the first author of this contribution). The three FITS files have comprehensive headers provided by all subsystems: ICS, VIS channel, NIR channel, acquisition and guiding and interlocks computers, front-end and calibration units controllers, Fabry-Pérot etalons, coudé room, telescope, dome and weather. All of them are connected to the ICS through a number of interfaces with their respective protocols.
4. *First pipeline*. A few seconds after the end of the observing block and spectra readout, CARACAL (CA[RMENES] Reduction And CALibration software[7]) automatically makes the dark/bias correction, order tracing, flat-relative optimal extraction[8] and wavelength calibration[9] of the VIS and NIR spectra and generates four processed files: two for each channel, one for each fibre in the field of view of the acquisition and guiding system. During standard operation at night, CARMENES simultaneously gets light from a target in the first fibre and the corresponding Fabry-Pérot etalon in the second fibre. However, it can instead get light from the sky in the second fibre when the target is faint, or light from U-Ne, Th-Ne and U-Ar hollow cathode lamps or a flat-field halogen lamp in one or two fibres for calibration. The nine files (five raw, four processed) are stored temporarily in the corresponding night folder in a disc partition of the main ICS computer.
5. *Observatory repository and data retrieval by principal investigator*. On the following morning, all files in the previous night folder are copied to a repository common to all instruments at the CAHA Observatory. Besides, the (public) calibration and (private) science data assigned to a certain programme are duplicated in a password-protected FTP site. When the duplication is finished, if this is the first time that CAHA Archive registers files from this principal investigator (not only this program), an automatic email is sent from caha.es with the instructions for retrieving all his the data via FTP.
6. *Guaranteed Time Observations Data Archive*. The previous five steps are common to open time programmes, director discretional time and guaranteed time observations (GTO). For GTO, the data flow continues with three more steps. The first of them is the set up of a data server located at the Centro de Astrobiología in Madrid that makes available all raw and processed data to the members of the consortium through a friendly web browser interface. The server makes an automatic daily copy of the FTP data and is virtual observatory compliant.
7. *Second pipeline*. Also an exclusive GTO step, SERVAL is a second pipeline that runs from the Institut für Astronomie Göttingen and computes precise radial-velocity measurements of the 300 monitored M dwarfs after telluric masking and proper échelle order weighting (CARACAL provides a rough estimate of the radial velocity based on a comparison with a high-resolution synthetic stellar model of low effective temperature and main-sequence gravity). SERVAL also measures the pseudo-equivalent width of Hα λ6562.8 Å, and has the power to implement new modules that measure other spectral activity indicators (indices of the calcium triplet, rotational velocities, emission lines in the near infrared, etc.).
8. *Visualisation tool*. The output of SERVAL is injected into RADAR, the RAdial velocity Data ARchive at the Max-Planck-Institut für Astronomie in Heidelberg. RADAR is a web-based tool to work with data generated by SERVAL. It provides users with standard and customisable sets of visualisation and analysis procedures with the final aim of detecting radial-velocity exoplanet signals and disentangling them from stellar activity. Closing the loop, RADAR (and SERVAL) needs some of the parameters originally tabulated by Carmencita.

2. SCHEDULING SOFTWARE

CAST is a standalone scheduler focused on translating the survey strategy into a detailed schedule (see Figure 2) for the achievement of the optimization goals. Specifically, CAST has to schedule thousands of observations related to the targets according to several hard constraints[3] (e.g., object elevation, object priority, Moon distance, pointing restrictions, exposition time, weather conditions). Some of these restrictions can be calculated in advance, but there are others that can only be computed in real-time (e.g., weather conditions, integration time). For this reason, CAST combines two scheduling strategies: off-line and on-line[10]. The off-line strategy provides a schedule of the observations without considering the constraints that cannot be predicted, and contains two different types of planning: (1) long-term schedule, with duration of several months; and (2) mid-term schedule, with the observations for a specific night. On the other hand, the on-line strategy repairs the off-line schedule in consonance with the constraints that must be calculated on the fly[11] and contains a short-term scheduler that reacts to unexpected situations by adapting the previously computed mid-term plan.

To fulfil this behavior, scheduling based on Artificial Intelligence (AI) techniques[12] is used in CAST to increase the efficiency of telescope operations, which will represent an important benefit in terms of scientific return and operational costs. The optimization goals of CAST are focused on maximizing the observing time of the night and the observation of the most suitable targets according to scientific criteria (e.g., target priority, target culmination, target magnitude).

The long-term and mid-term schedulers are not time-critical, for this reason, they are based on Genetic Algorithms (Gas)[13], which are a computationally expensive AI approach focused on emulating natural evolution by means of combining potential solutions using selection, combination and mutation operators[14]. The solution obtained with these algorithms highly optimizes the objectives defined in the problem. In CARMENES, these objectives are focused on fulfilling the hard constraints, maximizing the observing time of the night and performing the observation of the most suitable targets according to scientific criteria (e.g., target priority, target culmination, target magnitude). Optimization process was described in detail in previous contributions to SPIE[3].

Unlike the long-term and mid-term schedulers, the short-term scheduler is time critical because it must schedule an observation in a very short time (less than 5 seconds). For this reason, it uses astronomy-based heuristics[15] with the aim of avoiding intensive calculations by repairing the night schedule obtained by the mid-term scheduler.

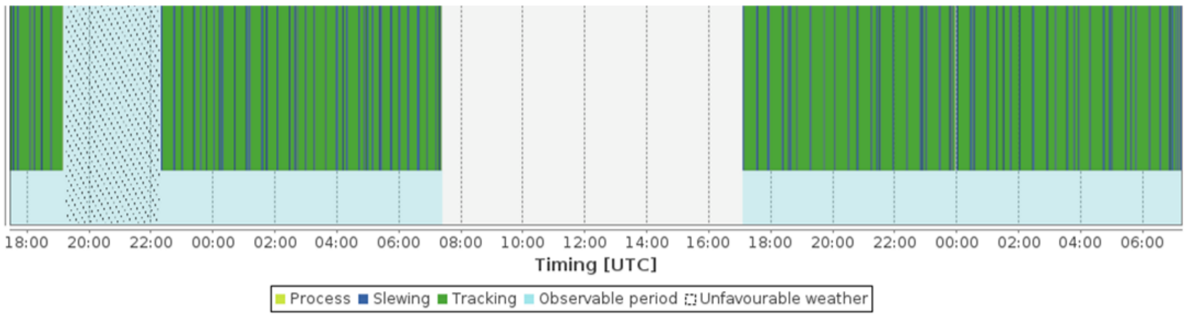


Figure 2: Night schedule for the CARMENES instrument

3. INSTRUMENT CONTROL SYSTEM

The ICS is the central software application in the CARMENES control layer. It is based on a modular architecture and a high level of abstraction design motivated by the heterogeneity of the different subsystems. Master/slave model architecture is used to build the control layer, where the ICS acts as a master and almost every other subsystem is a slave. The ICS acts as a slave only for the User Interface subsystem, which controls and monitors the ICS functionalities.

The instrument subsystems are abstracted into the ICS with a logic representation (see Figure 3). Each subsystem is divided into specific functionalities, such as data representation, actions and events. The subsystem data (e.g., temperatures, encoder positions) are reported periodically to the ICS using its communication protocol and are stored into a central database that conforms a pool of updated data for all the parameters of the system (like a snapshot).

Events (or contingencies) can arise during a nominal subsystem execution. An event is something that happens to the subsystem that generates a notification to the user or the execution of actions. The designed actions are encapsulated in a common API and the ICS triggers them using a common interface and without any dependence with the communication protocol. The overall system operation is, finally, handled by events that trigger predefined actions. These actions change the status of one or more components to reach the required configuration or response. Actions can be also triggered by an operator using any of the interface options (graphical or simple scripting) when the system is not running in the automatic control mode.

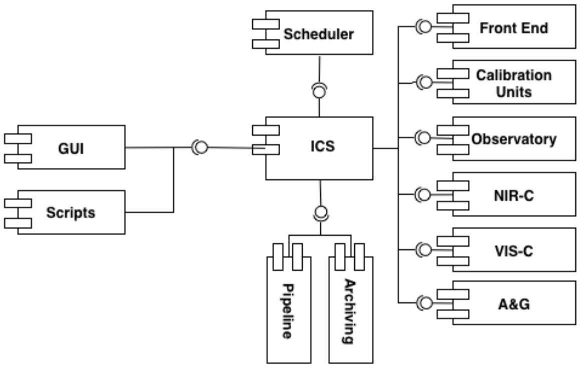


Figure 3: CARMENES logic components architecture.

3.1 Control layers

The ICS was designed using the Layers design pattern, which separates application functionalities into distinct levels of abstractions by decomposing complex problems into smaller and more manageable ones. The architecture is split in the following layers (Figure 4):

- **Operating System Layer:** This is the abstract layer that interacts with the operating system. It provides the necessary functionalities to manage threads, semaphores, mutex, file systems, etc.
- **Third-Party Libraries Layer:** It contains the libraries used from third party developers.
- **Modules Layer:** This layer does not process any data. Its aim is to manage a large amount of information by encapsulating it in data structures, which are grouped into modules.
- **Procedures Layer:** This layer defines all processes necessary to manage data and actions.
- **Subsystems Layer:** It contains the subsystems abstraction.
- **Communication Layer:** It contains all the protocols to communicate with the subsystems.
- **Interface Layer:** It defines all communication APIs to interact with the subsystems, modules and procedures.

Most of these layers, except the Operating System and the Third Party Libraries, are specifically designed for the CARMENES instrument. The ICS core functionalities are implemented in the Modules layer. Each module is designed to accomplish only one functionality and contains everything necessary to fulfill its purpose. This design reduces coupling and increases readability and maintainability.

The CARMENES control layer is composed by a heterogeneous set of subsystems at both physical and logical levels. Different physical implementations are used to run the low-level control software (i.e., microcontrollers, PCs) and some of them do not support using communication protocols that require high processing capability. This lack of homogeneity motivated the design of the ICS to support any communication protocol using a communication layer that hides the protocol implementation constrains. The ICS uses the Internet Communication Engine[*] (ICE) middleware as the base communication framework. ICE provides a complete solution to communicate different distributed subsystems that span multiple operating systems and programming languages. In addition to ICE, two more protocols are also implemented: Epics[**], used to interface with the telescope and dome control software; and a TCP/IP-based custom protocol (called CARMENES protocol), used to interface with the NIR and VIS channels, the front-end, and the calibration units.

---

[*] http://www.zeroc.com

[**] http://www.aps.anl.gov/epics/

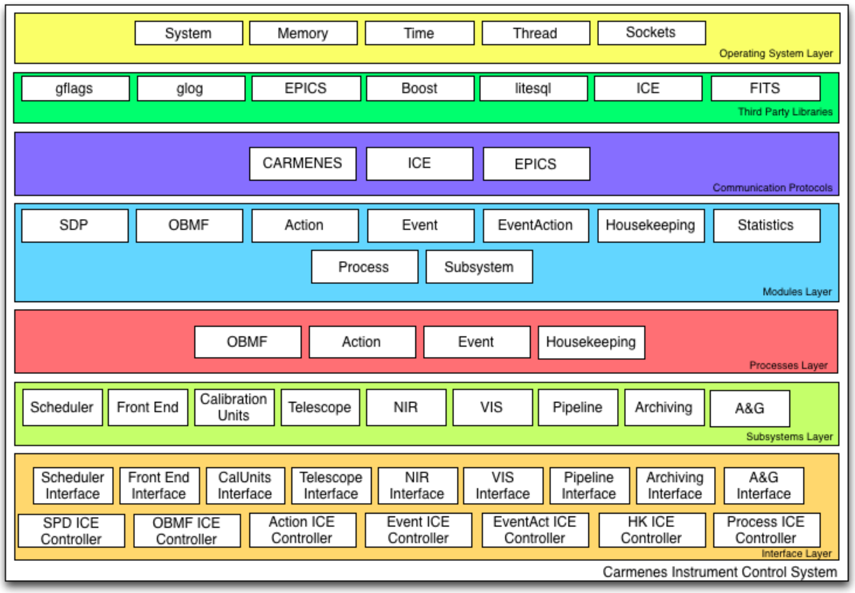
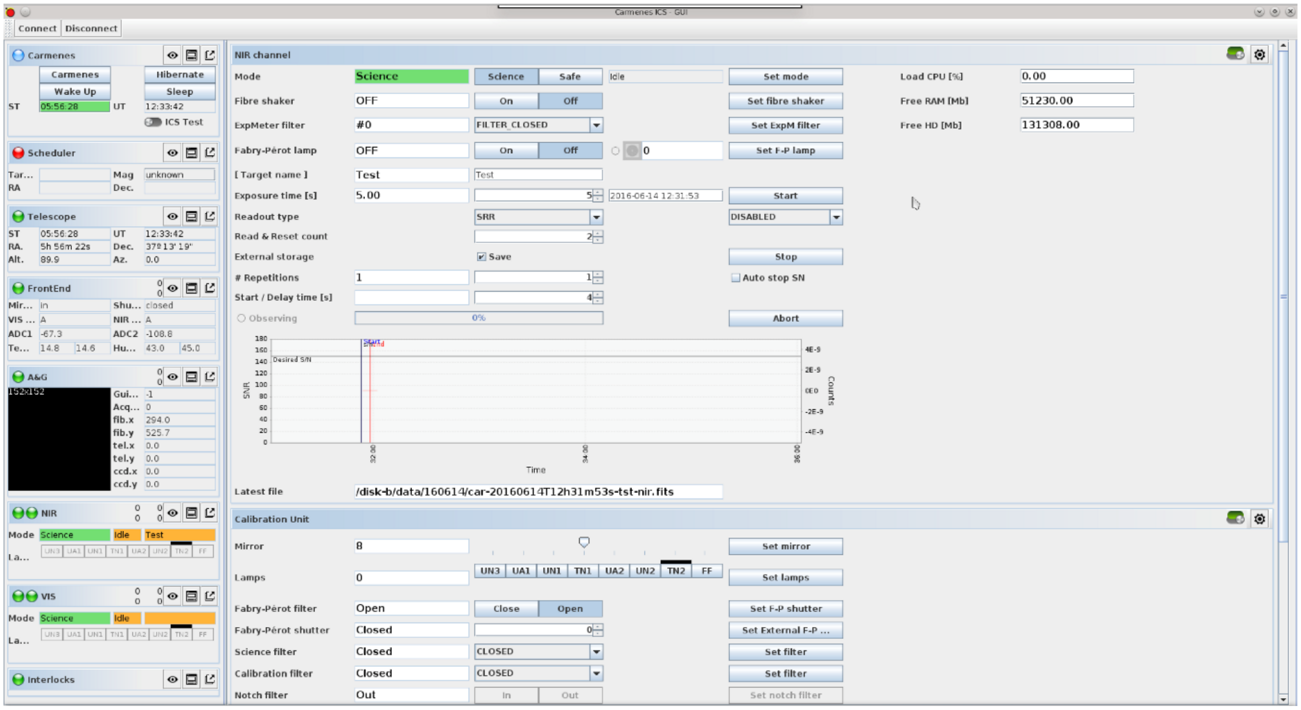


Figure 4: Layers and main modules of the CARMENES ICS.



ST 05:56:28 UT 12:33:42

Tar... Mag unknown
RA Dec.

ST 05:56:28 UT 12:33:42
RA. 5h 56m 22s Dec. 37º 13' 19"
Alt. 89.9 Az. 0.0

Mir... in Shu... closed
VIS ... A NIR ... A
ADC1 -67.3 ADC2 -108.8
Te... 14.8 14.6 Hu... 43.0 45.0

152x152
Gui... -1
Acq... 0
fib.x 294.0
fib.y 525.7
tel.x 0.0
tel.y 0.0
ccd.x 0.0
ccd.y 0.0

NIR: Mode Science Idle Test
VIS: Mode Science Idle

Mode: Science — Idle
Fibre shaker: OFF
ExpMeter filter: #0 — FILTER_CLOSED
Fabry-Pérot lamp: OFF — 0
[Target name]: Test — Test
Exposure time [s]: 5.00 — 5 — 2016-06-14 12:31:53
Readout type: SRR — DISABLED
Read & Reset count: 2
External storage: ☑ Save
Repetitions: 1 — 1 — ☐ Auto stop SN
Start / Delay time [s]: 4
○ Observing 0%
Latest file: /disk-b/data/160614/car-20160614T12h31m53s-tst-nir.fits

Load CPU [%]: 0.00
Free RAM [Mb]: 51230.00
Free HD [Mb]: 131308.00

Mirror: 8
Lamps: 0
Fabry-Pérot filter: Open
Fabry-Pérot shutter: Closed — 0
Science filter: Closed — CLOSED
Calibration filter: Closed — CLOSED
Notch filter: Out

Figure 5: CARMENES GUI snapshot (NIR control panel)

4. CARMENES GUI

The CARMENES GUI (Figure 5) has two main features: monitoring all the instrument subsystems and controlling them in case CARMENES operation mode is running[3]. Other operation modes, like the engineering control of all subsystems, are also supported by the GUI and are routinely used by the operators for very specific commissioning or maintenance tasks.

The GUI is implemented in Java and is based on JGoodies libraries[*]. The ICE framework is used to communicate with the ICS and subsystem monitoring is carried out using ICE features and using the publisher/subscriber design pattern. The ICS publishes the internal variables and the GUI is subscribed to all of them and shows their values to the operator. Every change in the ICS variables (internal data or subsystem data) is directly notified to the GUI and, thus, to the operator, without the need to refresh the GUI. A real-time feedback on subsystem ICE command execution is also received and visualized by the operator, including exceptions and the associated error messages.

Validation of variables is not carried out by the GUI, but by the ICS and subsystems.

The GUI architecture for such a distributed control system (as illustrated in Figure 3) was designed as an independent module in order to gain in maintainability, scalability, etc. However, the integration of the GUI and the ICS would have simplified the implementation and the validation procedures.

5. NIR CHANNEL CONTROL

Figure 6 shows the general structure of the CARMENES NIR channel, focused on elements that require some type of control made by electronics or software.

The working temperature of this channel (140 K) added complexity to the design of the NIR control system, as made the development of an innovative cooling system necessary, with deep implications not only in the opto-mechanics but also in the control electronics. Several control loops were implemented in order to reach the strict requirements in temperature stability (±0.07 K in 24 hours; ±0.01 K (goal)), and the performance of the system as a whole is monitored through a collection of several tenths of temperature sensors located in different points of every instrument subsystem. Pressure is another critical variable, as both the optics and the detector must operate in high vacuum conditions.

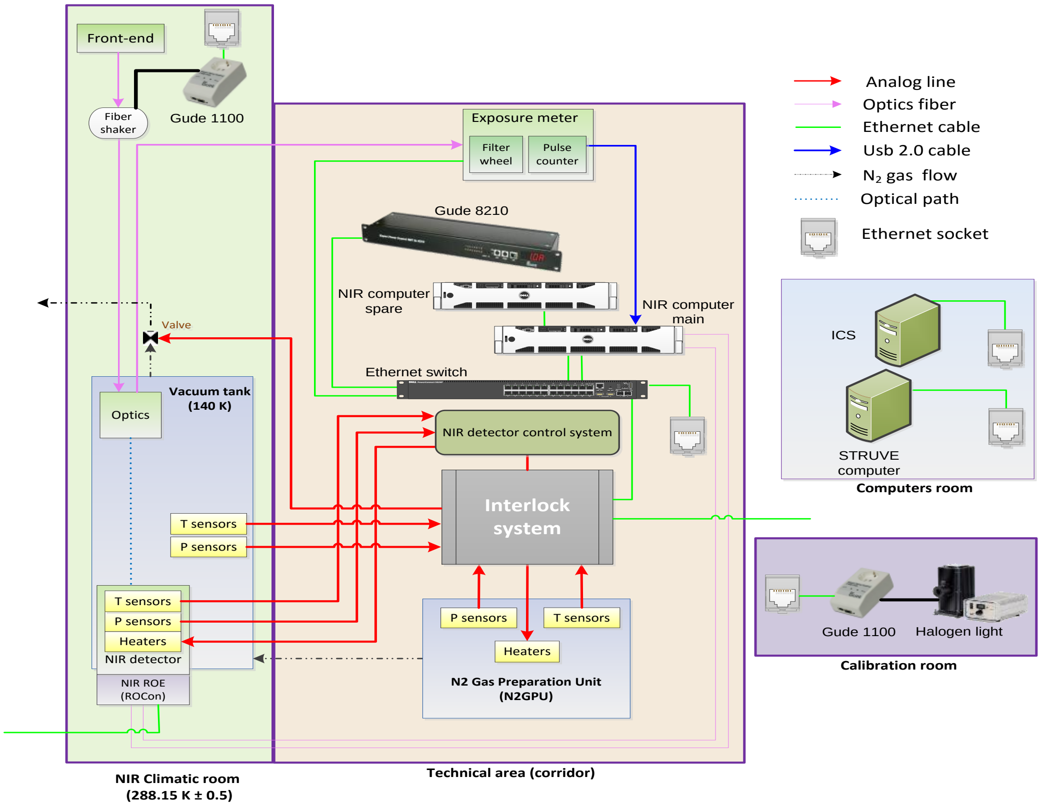


Figure 6: General description of electronics and software of NIR channel

There are four relevant areas for NIR channel at CAHA:

---

[*] http://www.jgoodies.com/

a) *NIR climatic room*. In this place will be located the Vacuum tank (VT), optics, sensors, NIR detector, NIR detector ROE, Front end (interface with the telescope), and fiber shaker (FS).

b) *Technical area* (*corridor*). It stores the main of NIR elements: Nitrogen gas preparation unit (N2GPU), Interlock System (IS), NIR detector control system, Ethernet switch, NIR spare computer, NIR main computer and Exposure meter (EM).

c) *Computers room*. It contains the central computer of CARMENES (called ICS) and the IS computer (called STRUVE running ScadaBR software).

d) *Calibration room*. This area has the last of NIR element: Halogen light (HL) that can be powered on/off remotely using GUDE 1100.

Areas **a)** and **b)** are contiguous in order to reduce the cable length.

The main beam from the front-end (interface with the telescope) is guided using an optic fibre to the vacuum tank (VT). In order to increase the beam stability, a fiber shaker (FS) is used. FS can be powered on/off by using a remote power module GUDE 1100. The optics, temperature sensors and NIR detector are inside the vacuum tank. The optics is in charge of providing the scientific beam to the NIR detector and to the fiber of the exposure meter (EM), which collects the zero order from the echelle. The EM has several elements that include a filter wheel and a set of electronic modules that amplify and count the photon flux. The NIR computer uses the final photon count to calculate the best time of exposition for the selected stellar object and observing conditions. The EM filter wheel and electronics modules can be remotely powered on/off using a remote power module GUDE 8210.

The NIR detector system is in charge of obtaining spectral images and sending them to the NIR computer. It is composed by a NIR detector; a NIR read out electronics (ROE), a temperature control system, a PCI board (OPTPCI) and the GEIRS software[16]. The data obtained by the NIR detector is managed by the OPTPCI board that is plugged into the NIR computer. The whole NIR detector system is controlled by GEIRS. The working temperature of the detector is lower than in the rest of the instrument (around 85 K), and is reached using a conventional cryostat with liquid nitrogen. The temperature control is achieved through several control loops in different levels of the detector cryostat, managed by two PID controllers (JUMO Imago 500 and Lakeshore 336).

As stated above, the optics in the NIR channel work in a high vacuum and low temperature (140 K) environment. In order to reach this condition, a VT and a cooling system (CS) are required. The VT has a set of 24 temperature Lakeshore DT670D-CU diode sensors distributed all over the optical bench and radiation shield. The 24 sensor cables are led to three concentrators (specifically designed PCB's) which output a unique cable to the connectors in the VT flange. Once outside the VT, the signals from the 24 temperature sensors are read through three temperature monitors Lakeshore 218S. In order to shorten the time between tests during the AIV phase, 16 heater boxes with thermostats were installed in different points of the optical bench and radiation shield. This reduces the time to get from the working temperature of 140 K to ambient temperature from several weeks to less than three days.

The main component of the cooling system is the N_2 gas preparation unit (N2GPU), which is housed in a secondary vacuum tank. This subsystem takes liquid nitrogen from an external bottle and produces highly-stabilized nitrogen gas. This nitrogen gas is the refrigerant used by the cooling system to cool the instrument and stabilize its temperature at 140 K. The N2GPU makes this phase conversion and stabilization in temperature in three different stages, each of them controlled by a PID controller JUMO dTron 316. As in the case of the main VT, the vacuum in the secondary tank and the temperature values in the three stages are monitored by different controllers.

The control and manipulation of the different active elements of the instrument, such as valves and pumps, is achieved through the alarms and interlocks system (IS). This is an electronic system based on Mitsubishi FX series PLC modules, which communicates with all the data controllers and monitors stated above and centralizes the data monitoring and control. The IS is originally a security device, programmed on base of algorithms designed to avoid the execution of certain actions under specific conditions. It also manages the alarms and warnings associated to any error, malfunction or dangerous situation, and it is also responsible of the initialization, stop, maintenance, supervision, contingency and recovery of the devices that is in charge of. The IS has its own computer (STRUVE) for controlling and data visualization. The IS is not specific to the NIR channel, it is conceived as a global subsystem. The interface between this IS and the user consists of a touch panel (Hakko Monitouch V8) from where all data can be viewed and the state of all the active elements can be modified.

Figure 7 shows the distribution of all the controllers, monitors and elements in the rack cabinet used for the NIR channel electronics.

NIR software, running in NIR computers, sends all collected data (coming from GEIRS and EM) to the ICS using an Ethernet link and a custom protocol. The halogen light will be used as source of an etalon to calibrate the NIR channel. It can be powered on/off by using a second remote power module GUDE 1100.

Finally, an Ethernet switch connects all elements of the NIR channel with Ethernet interface and it allows communicating with ICS and with external and external network of CAHA.

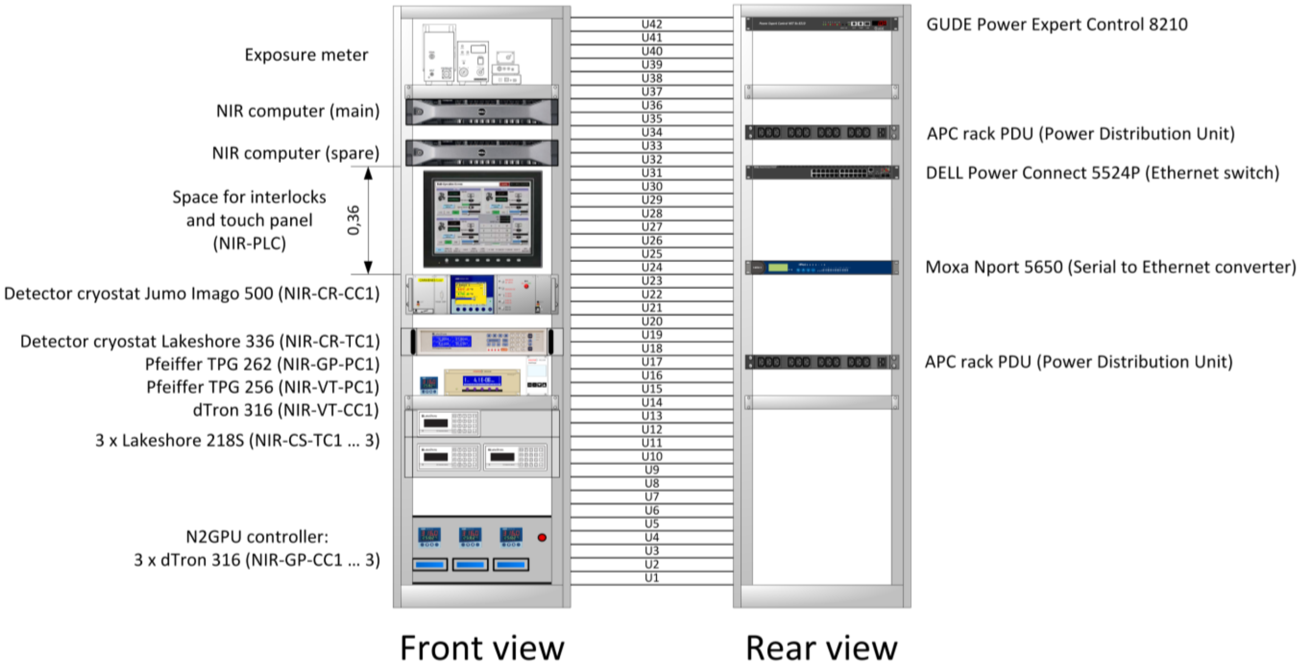


Figure 7: Distribution of the electronics elements in the NIR cabinet.

5.1 NIR SW main features

NIR software, implements all the actions required to obtain an image from detector following the observing parameters set by ICS and send it back to ICS. More in detail, it is in charge of:

- Subsystem control: GEIRS, FS, HL, FW and PC.
- System and subsystem on/off
- System operation mode change
- Communication protocol management
- Contingency and recovery manager
- Logging of all activities
- Scheduling of recurrent tasks
- Time synchronization with CAHA time server

5.2 Developing tools

The software (including engineering graphical interface) has been built mainly using in Java language and Eclipse developing environment. C++ language has been used to create Java-compatible libraries to manage some elements of

EM. A custom set of tools has been developed for: automatic documentation generation, requirements management, automatic production of user graphical interface and script suit for testing purposes.

6. VIS-CHANNEL CONTROL

The VIS channel computer controls the CCD camera of the visual channel of CARMENES. The controller of the camera is situated at the spectrograph tank and is connected via data fibres to a proprietary PCIe-card plugged in the control computer.

The computer also controls the exposure meter of the VIS channel and registers its measured counter values. The counter of the exposure meter is connected to a USB port of the computer.

The fibre shaker and the lamp of the etalon used by the calibration unit are switched via PDUs. The PDUs are accessed following SNMP via Ethernet connection.

6.1 VIS SW main features

The software performs exposures with the CCD camera producing images in FITS data format.

It configures the counter of the exposure meter and controls its filter wheel. It protects the PMT of the exposure meter by automatically selecting the filter if the incoming flux is too high. Therefore it continuously reads the counter values when the exposure meter is powered on. During CCD exposures it makes them available via the interface.

The fibre shaker and the etalon halogen lamp are switched by using system calls provided by the repository package 'net-snmp'.

The software acts as a gateway to the visual channel by running as a server providing a tcp/ip interface.

6.2 Software Design

Two tcp/ip servers are running concurrently. The main server provides the interface to the whole spectrograph, mainly accessed by the ICS. The second server is exclusively dedicated to handle the exposure meter and its filter wheel. Usually its sole client is the main server.

Both servers are controlled by the init sytem 'systemd' used in OpenSuse.

The main server programme runs two threads in parallel:

- main thread: It provides the interface and the command parsing. Additionally, immediately responding tasks are fulfilled like switching the PDUs. There is no need for asynchronous hardware handling.
- exposure thread: This thread uses a stand-alone programme that controls the camera, takes exposures, and stores image files in a ring buffer.

The second server programme has a similar structure:

- main thread: It provides the interface and the command parsing and switches the power.
- exposure thread: This thread runs when the exposure meter is powered on. It takes care of the handling of the counter and logging of its measured values.

The realized structure allows the immediate answering and acknowledging of commands, while in parallel hardware dependent functions are run in their own thread and at their own step rate.

'Make' is used to compile the C source code and link the executable binaries.

7. FRONT-END CONTROL

The front-end of CARMENES (Figure 8) is mounted at the Cassegrain focus of the 3.5m telescope. It provides the possibility to feed the focused light collected by the telescope into the fibres of both the channels of the spectrograph. If not activated, the light is passed through to the PMAS instrument.

Dedicated electronics and software provides an interface to control the optical and mechanical parts or devices needed for the observation. The electronics housing mounted at the front-end also contains the acquisition and guiding computer that controls the guiding camera located in the front-end. The proximity of the computer gives the opportunity to completely control the firmware of the front-end-electronics and reprogram it remotely. Apart from that the systems are independent of each other.

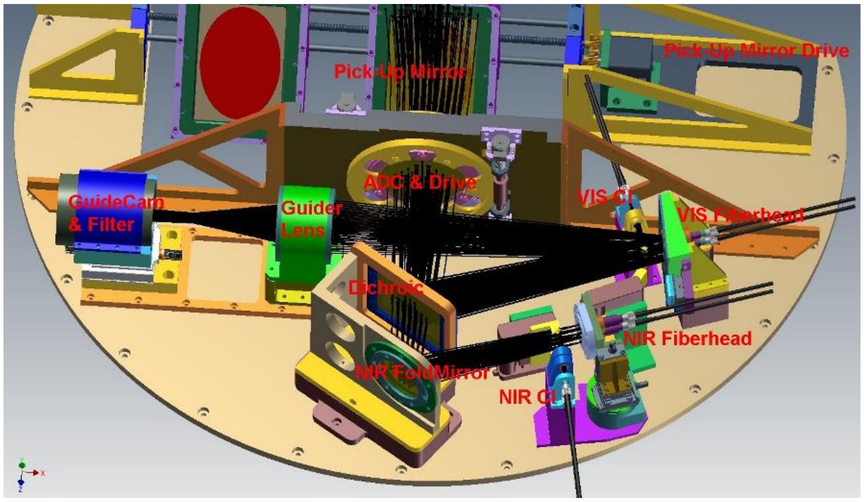


Figure 8: Front-End, 3d view of inside with labels

7.1 Devices and sensors to be controlled in the front-end

The electronics of the front-end (Figure 9), consisting of an Arduino Mega micro controller board, a dedicated PCB for the main electronics, and a small utility board, handles the following devices:

- It moves the pick-up mirror between two positions, in or out
- It rotates the ADC prisms between -180° to 180°
- It moves the shutter in front of the VIS fibres between two positions
- It moves mirrors in front of the fibres from the NIR and VIS calibration units between four positions: no calibration light in both science fibres, calibration light reflected into the first fibre, calibration light in both fibres, and calibration light in the second fibre
- It provides the temperature and relative humidity at two positions in the front-end: one near the fibre inputs and one near the guiding camera
- It provides the temperature in its housing

The pickup mirror is driven by a stepper motor via a spindle. The two discrete positions of the pickup mirror (in and out) are indicated by Hall sensors. The ADC prisms are also rotated by stepper motors via a worm wheel to angles from -180° to 180°. A reference position is indicated by a Hall sensor for each prism. On each motor axis a rotary encoder is mounted to determine the actual positions of the prisms.

All three stepper-motors and both encoders are directly controlled by the micro controller.

The mirrors in front of the fibres are moved by compact micro-translation stages M-112 from 'Physik Instrumente GmbH' (PI). In conjunction with the matching controller C-863, the stages can be moved to pre-defined positions – repeatability accuracy of 0.0002 mm – by calling previously programmed macros via an RS 232 connection. TTL output ports of the controller signal arrival at these positions. Additionally a Hall sensor indicates the home position ('no calibration light in both fibres').

The shutter blade is moved by a fast miniature linear motor stage M-661 from PI. The matching controller C-867 has the same interface as the C-863 controller. Both positions are indicated by Hall sensors.

The communication with the front-end is TCP/IP based. The controller, equipped with an Ethernet board extension, acts as a server, mainly for the ICS. It allows three concurrent client connections. The interface is always capable of receiving commands. Every request will be acknowledged or repelled immediately.

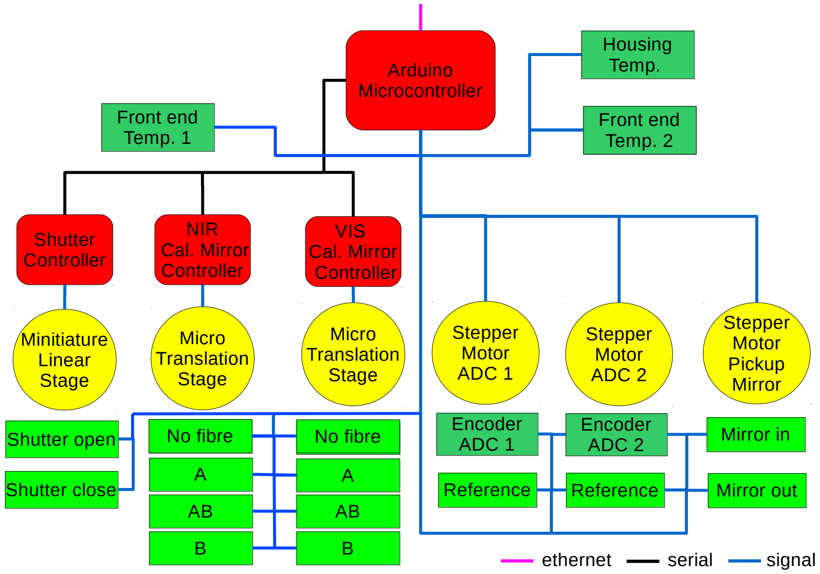


Figure 9: Front-end control electronics design

7.2 Software implementation

The software was developed in the standard Arduino IDE (www.arduino.cc) in the Linux environment of the computer for the acquisition and guiding system. It can be updated and recompiled remotely. Transfer into the flash memory of the micro-controller may take place via the permanent USB connection.

8. ACQUISITION AND GUIDING SYSTEM CONTROL

A Starlight Xpress SVX-H16 camera is used for acquisition and guiding. It may be switched on and off via a PDU connected to the ethernet. It looks onto a mirror in front of the VIS fibre head which has two holes for the science and the calibration fibre at a distance of 88” with a diameter of 1.5”. The camera provides a field of view with a diameter of 3'. Binning 2x2 yields a pixel size of 0.18” and read-out times less than 1s. It is connected via USB to a dedicated computer situated in the front-end electronics box.

8.1 Global Software design

A daemon process provides TCP/IP based accessibility. It acts as a server for the ICS. It allows starting, stopping, and monitoring of the acquisition and guiding process. The acquisition and guiding itself is performed autonomously by the system.

The server programme runs two threads in parallel: the interface and house keeping thread and the guiding thread.

This design allows the immediate answering and acknowledging of commands, while concurrently the imaging and guiding process runs independently following its own timeline.

A stand-alone programme controls the camera, takes exposures, and stores image files in a ring buffer. It is used by the guiding thread to get camera images.

8.2 Acquisition and Guiding

The A&G system allows the acquisition of a target into the science fibre of the CARMENES spectrograph and subsequent guiding. After finishing the absolute positioning of the telescope an acquisition exposure is taken. Then the A&G system positions a selected target detectable in the FOV of the guiding camera into the fibre aperture and holds the telescope position with an accuracy depending on the seeing and the properties of the telescope. During this process the A&G software automatically takes further images with optimized exposure times determined by the signal in the images.

After a configurable period the images are stacked and a new telescope position correction is determined. On one hand the position of the target itself in the fibre hole is determined. On the other hand the positions of all field stars that were detected in the acquisition image are determined and a relative position of the target is computed. A weighted average with configurable factors of both positions provides the telescope offset. If only one kind of position is determinable it is used solely.

8.3 Software implementation

The software was developed in C in the Linux environment OpenSuse. Make was used for compilation and link. The daemon process is integrated into the operating system via systemd.

9. INTERLOCKS SOFTWARE

Interlocks is the subsystem responsible for ensuring the extremely high stability in the environmental conditions required by CARMENES science project and implementing the safety measures to avoid any personnel injury or damage to any important element of the instrument. To meet these requirements Interlocks is based in two units, one low-level control which interacts directly with devices and hardware and other high-level control oriented to facilitate the supervision and data logging.

The so-called low-level control is based on several Programmable Logic Controllers (PLC) connected to different types of devices including Cryogenic Temperature Controllers, Vacuum Controllers, Turbopumps and High Precision Thermoregulators, between others. These PLC systems are responsible for monitoring the proper operation of the devices set and respond appropriately if an abnormal or dangerous situation arises.

Furthermore, the high-level control unit is composed by a a Supervisory Control And Data Acquisition software application, also known by its acronym SCADA, and a daemon service. The SCADA gathers and stores information from different controllers in the Interlocks subsystem, offers graphical views to simplify the user interaction, generates useful reports and historical graphs to enhance the performance, and stability, and detects any abnormal or risky situation contacting the technical staff in case any actuation is required. The daemon service, running on the same computer as the SCADA, is developed specifically as a data interface between the SCADA and other CARMENES subsystems that support the Subsystem Communication Protocol (SCP) used to communicate each other.

The SCADA system used for CARMENES is based on ScadaBR, an Open Source Software developed initially by Serotonin Technology Inc. and evolved by the ScadaBR community. Due to its open source nature, it has been possible to develop customized data sources not included into the ScadaBR software application by default. Two new data sources has been implemented, a fully configurable serial communication (available as datasource **“Custom Serial”** in the ScadaBR customized application) and a raw socket connection to a concrete host and port (available as datasource **“Socket Communication”** in the ScadaBR customized application). This SCADA software is hosted in a Apache Tomcat web server which receives HTTP requests from a web browser, processes the requests, reads from or writes to its own MySQL datasource, generates an HTTP response and sends it back to the web browser where the SCADA graphical interface is updated. The basic architecture of the ScadaBR application for CARMENES Interlocks is shown in the Figure 10.

For CARMENES Interlocks system, two different ScadaBR entities have been implemented, one of them for the Visible channel, known by the abbreviation VIS, and the other for the Near-Infrared channel, known by the abbreviation NIR. Several data sources have been configured into each of these entities to monitor the main controllers of the Interlocks System including: a Modbus IP communication with Mitsubishi Programmable Logic Controllers, Serial communication to Lakeshore Temperature and Pfeiffer Vacuum Controllers, Socket communication to Lakeshore 336 Temperature Controller and Modbus Serial communication to JUMO IMAGO 500 Mutichannel Process Controller. To avoid a degradation to the performance of the computer that hosts the ScadaBR entities, the majority of the values captured from these data sources are stored into the database during one month although their databases are backed up every day, so all the data values can be recovered if needed.

The SCADA tool provides an Application Programming Interface (API) based on standard protocol Simple Object Access Protocol (SOAP) that offers a simple communication interface to events and data handled by the SCADA entity. The daemon service gets access to these data and events through the API and interchange them with other CARMENES subsystems using the SCP protocol.

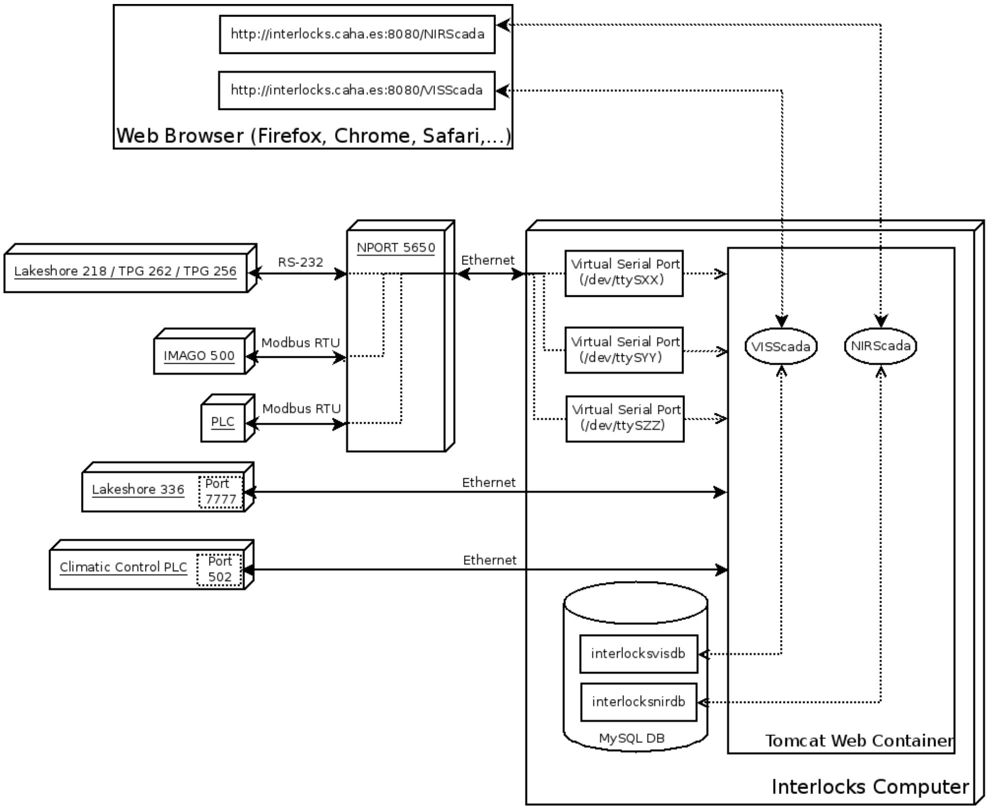


Figure 10: General architecture for Interlocks high-level control unit based on ScadaBR

10. TELESCOPE CONTROL: THE T_COMMAND SYSTEM

The t_command system is the interface that allows ICS to communicate with the telescope. This interface allows ICS having a unique and transparent way for accessing the telescope, regardless of the telescope internal control method. Thus, how the telescope is managed internally is transparent to ICS, and even if such control system is changed in the future, ICS will suffer no modifications at all.

10.1 Design and architecture

The system is designed to run into a separate computer, allowing any instrument requesting a telescope action and/or any status information to communicate with it through an encrypted ssh communication channel. Only allowed users within allowed computers can interact with t_command system. And t_command is the only service accessible by such privileged users at the computer where this interface runs.

The dedicated computer is a very simple virtual machine running within a VMware virtualization system, with high availability, inside a cluster of three servers. The computer is a 1 vCPU computer with 2 GB of RAM and 50 GB of disk space. The Operating System is OpenSuSE 13.1 (x86_64) and it has all the needed software: Tcl/Tk, Tcllib and the Oag library for the EPICS system environment. The system also has a very extensive log that can be accessed in the same way as when communicating with the telescope, and permits knowing which actions were taken in the past and their results, or any error that had happened.

For the secure ssh communication an asymmetric cryptography system is used, holding the instrumentation computer's user public key within the dedicated computer, so the access from the instrument computers (i.e. ICS) is carried out directly, without needing any password. Besides, all configuration necessary for accessing the telescope (i.e. environmental variables or access to EPICS system) is prepared on the dedicated computer, so the instrument computer has to know nothing about how to instruct the telescope but the commands allowed by the t_command system: t_dfocus, t_offset, t_posit, t_coord_system, t_tracking, t_request and t_log.

The way the communication with t_command works is by sending the desired telescope command through the secure ssh path to the dedicated t_command system computer. The system running inside that machine will, after some sanity checks, communicate with the telescope (at present, actions are achieved by contacting the telescope EPICS database) and will perform the requested action. T_command system will answer with a code (or a complete message if in debug mode) containing the operation result, so the caller system knows when the requested action is finished, the telescope status or if there was an error (and what error).

10.2 Benefits

The main benefit of t_command system is the abstraction layer this application offers to the instrument computer applications in their interactions with the telescope. But this layer is also a standardized and safe method for accessing the telescope system, preserving it from uses that could endanger it.

Finally, the way the system is designed gives other instrument the chance to use the same method in any of the main Calar Alto Observatory telescopes. In this way, the t_command system is already used not only with CARMENES, but also with the 2.2m PANIC instrument.

11. CONCLUSIONS

CARMENES is a new generation instrument installed at the Zeiss 3.5m Calar Alto Telescope and in operation since January 2016. The efficient and reliable operation control is a requirement for the instrument to maximize the scientific return. Such operational requirements resulted into strong constraints to define the control layer at the level of hardware and software, and all the subsystems that compose it. In this sense, the Instrument Control System (ICS) is focused on fulfilling the operation requirements and handling a heterogeneous group of subsystems in a coordinated manner. We have described the different subsystem control modules that imposed important constraints to end up with a complete suite of software providing a robust, optimized and reliable control. The first six months of operation[17] has shown the suitability of the implemented solution.

12. ACKNOWLEDGEMENTS

CARMENES is an instrument for the Centro Astronómico Hispano Alemán de Calar Alto (Almería, Spain). The observatory is operated jointly by the Max-Planck-Institut für Astronomie of the Max-Planck-Gesellschaft and the Instituto de Astrofísica de Andalucía of the Consejo Superior de Investigaciones Científicas. We acknowledge financial support from the Spanish Ministry of Economy and Competitiveness (MINECO) and the "Fondo Europeo de Desarrollo Regional" (FEDER) through grants AYA2012-39612-C03-01, AYA2011-30147-C03-02 and AYA2011-30147-C03-03.

REFERENCES

[1] Quirrenbach, A., Amado, P. J., Mandel, H. et al., "CARMENES: Calar Alto high-Resolution search for M dwarfs with Exoearths with a Near-infrared Echelle Spectrograph", Proc. SPIE 7735, E37 (2010).
[2] Quirrenbach, A., Amado, P. J., Caballero, J. A. et al., "CARMENES instrument overview", Proc. SPIE 9147, E1F (2014).
[3] García-Piquer, Á., Guàrdia, J., Colomé, J. et al., "CARMENES instrument control system and operational scheduler", Proc. SPIE 9152, E21 (2014).
[4] Caballero, J. A., Cortés-Contreras, M., Alonso-Floriano, F. J. et al., "CARMENES at PPVI. CARMENCITA Herbs and Spices to Help you Prepare a Genuine Target Sample", Protostars and Planets VI, 15-20 July 2013, Heidelberg, poster #2K020 (2013).
[5] Quirrenbach, A., Caballero, J. A., Amado, P. J. et al., "Manufacturing, Assembly, Integration and Verification of CARMENES and Preparation of its Input Catalogue", 18th Cambridge Workshop on Cool Stars, Stellar Systems, and the Sun, 8-14 Jun 2014, Lowell Observatory, USA, 897 (2015).

[6] Alonso-Floriano, F. J., Morales, J. C., Caballero, J. A. et al., “CARMENES input catalogue of M dwarfs. I. Low-resolution spectroscopy with CAFOS”, A&A, 577, A128 (2015).
[7] Zechmeister, M., Anglada-Escudé, G., Reiners, A., “Flat-relative optimal extraction. A quick and efficient algorithm for stabilised spectrographs”, A&A, 561, A59 (2014).
[8] Zechmeister, M., Zhao, J., Marvin, C. et al., “CARACAL – The CARMENES pipeline”, Extreme Precision Radial Velocity workshop, 5-8 July 2015, Yale University, USA, published on-line (2015).
[9] Bauer, F. F., Zechmeister, M., Reiners, A., “Calibrating echelle spectrographs with Fabry-Pérot etalons”, A&A, 581, A117 (2015).
[10] Rasconi, R., Policella, N., and Cesta, A., “Integrating Off-Line and On-Line Schedulers,” in [Proceedings of the 17th European Conference on Artificial Intelligence], Frontiers in Artificial Intelligence and Applications 141 , 845–846, IOS Press (2006).
[11] Rasconi, R., Policella, N., and Cesta, A., “Fix the Schedule or Solve Again? Comparing Constraint-Based Approaches to Schedule Execution,” in [Proceedings of the ICAPS Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems], 46–53 (2006).
[12] Donati, A., Reinhold, B., Martinez-Heras, J. A., and Policella, N., “Why Introducing Innovative Technology in Operations?,” in [The 12th International Conference on Space Operations], (2012).
[13] Holland, J. H., [Adaptation in natural and artificial systems], The University of Michigan Press, Ann Arbor (1975).
[14] Freitas, A. A., [Data Mining and Knowledge Discovery with Evolutionary Algorithms], Springer-Verlag New York, Inc., Secaucus, NJ, USA (2002).
[15] Giuliano, M. E., Rager, R., and Ferdous, N., “Towards a Heuristic for Scheduling the James Webb Space Telescope,” in [Proceedings of The International Conference on Automated Planning and Scheduling], 160–167, AAAI (2007).
[16] High Energy, Optical, and Infrared Detectors for Astronomy V. Proceedings of the SPIE, Volume 8453, article id. 84532E, 14 pp. (2012).
[17] Quirrenbach, A., Amado, P. J., Caballero, J. A. et al., "CARMENES: an overview six months after first light”, Proc. SPIE 9908, this volume