Design of the LCU for HARPS-N

Log book of the 2nd technical mission at La Palma

Date : 13.08.2017

From : M.Sarajlic

To : Observatoire de Genève

Reference :

HARPS-N LCU blok diagram

HARPS-N FEU and CU Control Electronics

LCU_hardware_connection_diagram_and_test_plan

LCU hardware connection diagram for ICS and test plan

RECORD OF REVISIONS

ISS/REV	Date	Modifications	Created/modified by
1/0	11.08.2017		M.Sarajlic

Table des matières

1 INTRODUCTION

The present report has for goal to list all activities and results during the technical mission from 11.08.2017 to 18.07.2017 on site at La Palma.

The objective of this mission is to implement the new LCU software onto the new LCU computer and test all functionalities. Finally, one observation night will be made with the new software. Ideally, we want to replace the old LCU with the new one after our mission.

2 HARDWARE

We have initially started to characterize the type of hardware used by the LCU and install all required drivers to be able to communicate and control through LabView software.

Here below is an updated list of hardware used with their LabView name and their hardware references. Most of the hardware come from the company Newport and so the controllers ESP301 as well. You can find so most of the drivers and firmware's on their site.

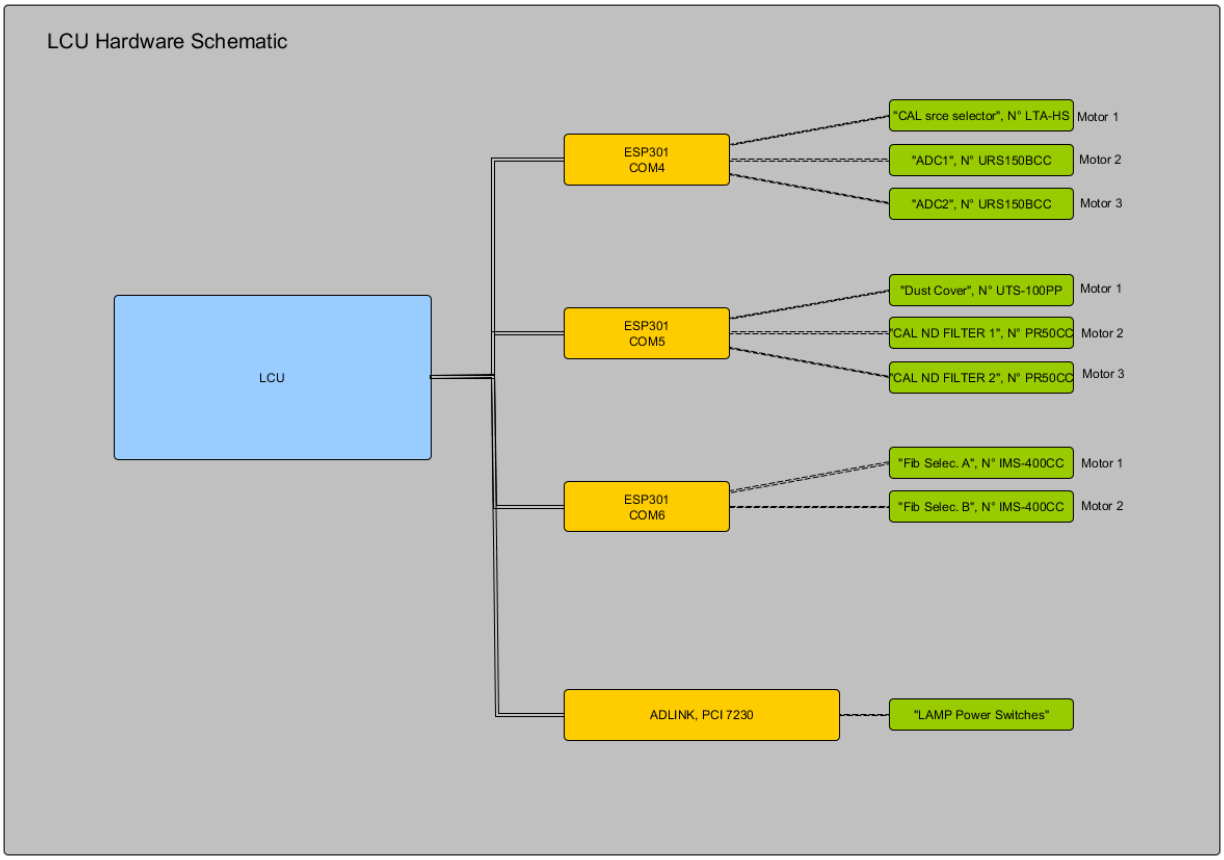


Figure 2-1 Schematic of the hardware on HARPS-N with its connection. The LCU comprise all instruments connected to the ESP301 drivers and get sensor values through the PCI-7230 card.

2.1 ESP301 controller

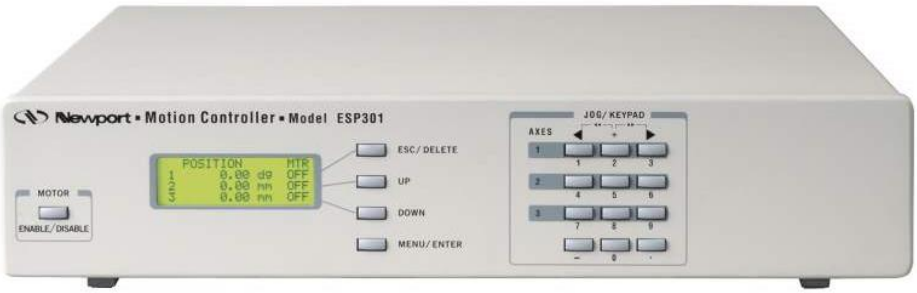


Figure 2-2 ESP301 controller from Newport

The most recent firmware of the ESP301 controller has not been installed as it is unknown if it may making crash the old LCU. This update can be done once the new LCU integrated.

The firmware correct mostly some **USB communication** which the actual case for the LCU.

2.2 ADLINK PCI 7230 card

The PCI card which is used to control the lamps is a standard card which has 16 digital inputs and outputs. Its role is to be a logic switch for the real electrical power supplies for the lamps. The PCI card is incorporated into the computer and connected directly to the motherboard.

- DO 0 , **Tungsten lamp output**
- DO 1 , **Thorium- Argon 1 lamp output**
- DO 2 , **Thorium- Argon 2 lamp output**

The lamp provides also 3 signals to the old LCU which apparently gives the state of the lamps.

- DI 0 , **Tungsten lamp input**
- DI 1 , **Thorium- Argon 1 input**
- DI 2 , **Thorium- Argon 2 input**

Unfortunately there is no spares of this card making the transfer to the new LCU complicated and has been transferred to the new PC each time some tests were performed.

The new LCU can control all lamps and also read all lamps status. All drivers have been installed into the new LCU computer (hardware drivers and LabView drivers)

It is clear now that a spare card is needed. One option would be to buy an identical one which will work with the current old LCU and the new one. The card price is 155.-CHF.

Figure 2-3 Current card used in the old LCU and for which the new LCU code was implemented. This card is installed directly onto the computer motherboard.

Angebot AN2696370822 / mail fr. 22.08.2017

Pos	ID code	Description	Qty (Back Order)	Unit Price	Amount
#1	PCI-7230G	PCI isolated Digital IO Card, 16/16 DIO	1 (-)	155,00	155,00
		PCI bus isolated Digital IO card, 16/16 isolated DIO, output darlington open collector max 40VDC 200mA sink current, input resistance 1k2 Ohm max24VDC, troughput 10kHz, isolation voltage 5kV, 37pin female D-type connector, 153*107mm, ADLink			
		more info @ : www.acceed.com/pci-7230			
		Country of Origin: Taiwan R.O.C., Custom Code: 8473 3020 000			

Figure 2-4 Quote got from ADLINK for this specified card

The second option would be to buy two other cards which are less professional but more adequate to their use (simple switches). National Instrument offer few of them.

NI USB-6501 (24 voies, 8,5 mA)

Figure 2-5 NI USB-6501 24 DI/O, USB 2.0 price is 120.- CHF

3 SOFTWARE

3.1 Hardware drivers

All hardware drivers can be found into the folder New LCU Drivers.

Here is the procedure to install hardware drivers for the ESP301 controllers :

1. Install ESP301 GUI_V2.0.0.3
2. Go into C:\Newport\Motion Control\ESP301\Bin\USB Driver\32-bit
3. Install the USB drivers if you are working onto a 32-bit OS (suggested)
4. Check in the device manager that the ESP301 is recognized

For the PCI ADLINK the PCIS-DASK v5.21 has to be installed and the computer restarted.

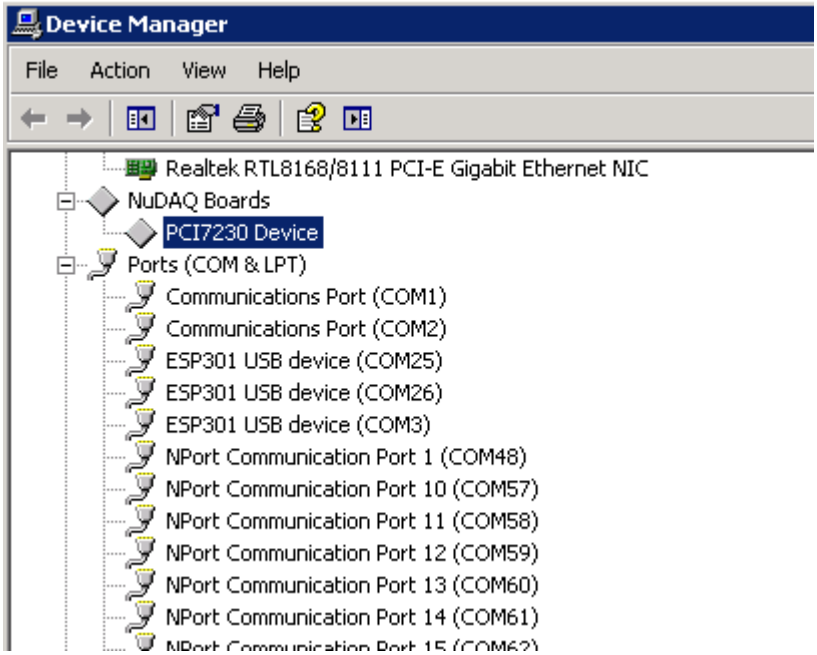


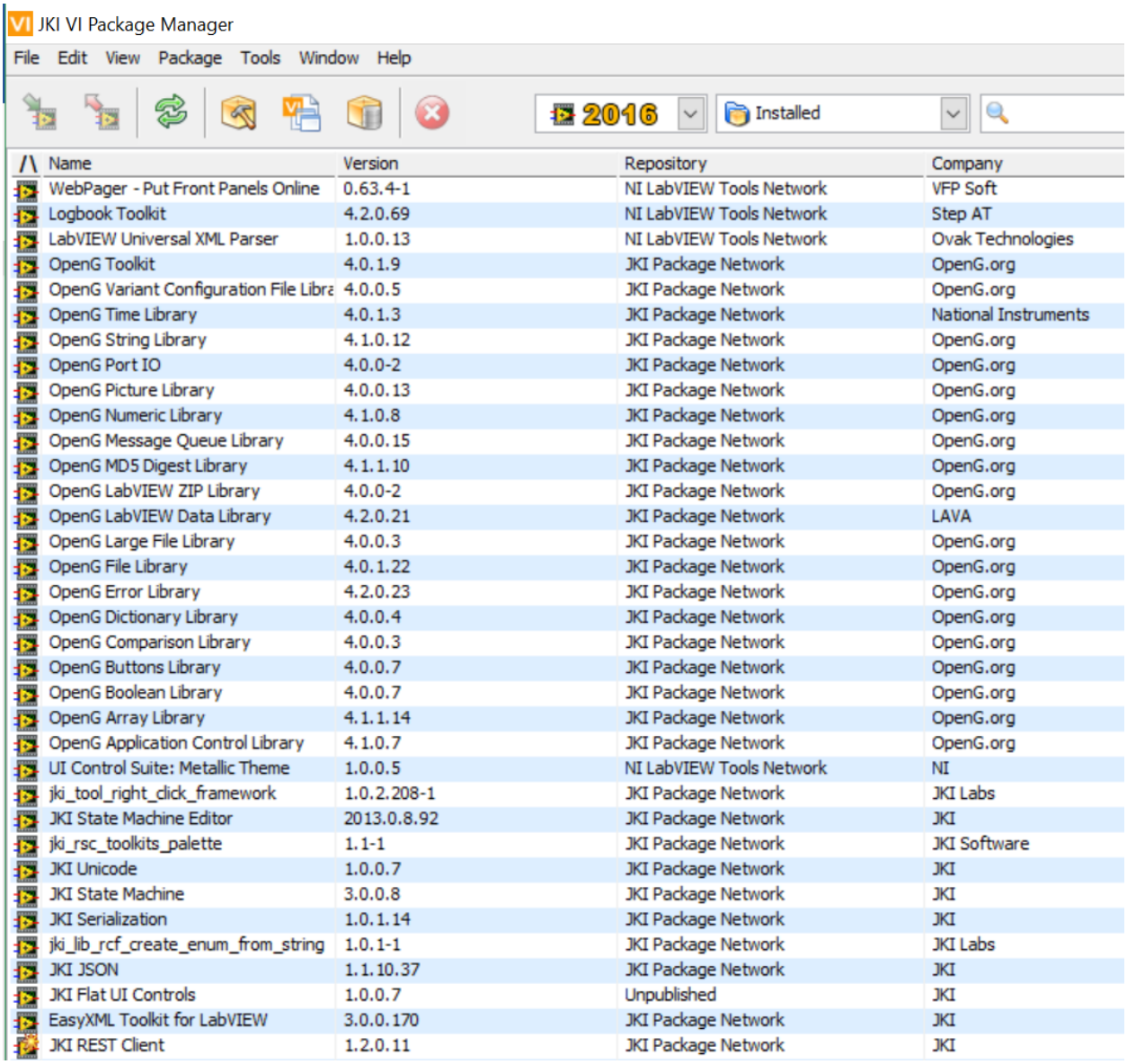
Figure 3-1 Old LCU device manager. We can see the PCI express card installed in the computer and the 3 COM ports (COM 25, COM 26 and COM 3) of the ESP301 driver.

3.2 LabView drivers

In this chapter, all LabView softwares including all addons and drivers are listed to be able to operate correctly the new LCU.

Firstly the LabView Professional Development System 2016 or more recent is required to be able to communicate via the web methods urls with the Sequencer. It is suggested that maximum of toolkits are selected during installation to ensure missing toolkits afterwards.

Once the LabView installed, install the JKI VI Package Manager (http://jki.net) which is a software allowing installing additional toolkits and LabView palettes. Here is the list of packages required. Ensure that all are installed with the same version or more recent.



Name	Version	Repository	Company
WebPager - Put Front Panels Online	0.63.4-1	NI LabVIEW Tools Network	VFP Soft
Logbook Toolkit	4.2.0.69	NI LabVIEW Tools Network	Step AT
LabVIEW Universal XML Parser	1.0.0.13	NI LabVIEW Tools Network	Ovak Technologies
OpenG Toolkit	4.0.1.9	JKI Package Network	OpenG.org
OpenG Variant Configuration File Libra	4.0.0.5	JKI Package Network	OpenG.org
OpenG Time Library	4.0.1.3	JKI Package Network	National Instruments
OpenG String Library	4.1.0.12	JKI Package Network	OpenG.org
OpenG Port IO	4.0.0-2	JKI Package Network	OpenG.org
OpenG Picture Library	4.0.0.13	JKI Package Network	OpenG.org
OpenG Numeric Library	4.1.0.8	JKI Package Network	OpenG.org
OpenG Message Queue Library	4.0.0.15	JKI Package Network	OpenG.org
OpenG MD5 Digest Library	4.1.1.10	JKI Package Network	OpenG.org
OpenG LabVIEW ZIP Library	4.0.0-2	JKI Package Network	OpenG.org
OpenG LabVIEW Data Library	4.2.0.21	JKI Package Network	LAVA
OpenG Large File Library	4.0.0.3	JKI Package Network	OpenG.org
OpenG File Library	4.0.1.22	JKI Package Network	OpenG.org
OpenG Error Library	4.2.0.23	JKI Package Network	OpenG.org
OpenG Dictionary Library	4.0.0.4	JKI Package Network	OpenG.org
OpenG Comparison Library	4.0.0.3	JKI Package Network	OpenG.org
OpenG Buttons Library	4.0.0.7	JKI Package Network	OpenG.org
OpenG Boolean Library	4.0.0.7	JKI Package Network	OpenG.org
OpenG Array Library	4.1.1.14	JKI Package Network	OpenG.org
OpenG Application Control Library	4.1.0.7	JKI Package Network	OpenG.org
UI Control Suite: Metallic Theme	1.0.0.5	NI LabVIEW Tools Network	NI
jki_tool_right_click_framework	1.0.2.208-1	JKI Package Network	JKI Labs
JKI State Machine Editor	2013.0.8.92	JKI Package Network	JKI
jki_rsc_toolkits_palette	1.1-1	JKI Package Network	JKI Software
JKI Unicode	1.0.0.7	JKI Package Network	JKI
JKI State Machine	3.0.0.8	JKI Package Network	JKI
JKI Serialization	1.0.1.14	JKI Package Network	JKI
jki_lib_rcf_create_enum_from_string	1.0.1-1	JKI Package Network	JKI Labs
JKI JSON	1.1.10.37	JKI Package Network	JKI
JKI Flat UI Controls	1.0.0.7	Unpublished	JKI
EasyXML Toolkit for LabVIEW	3.0.0.170	JKI Package Network	JKI
JKI REST Client	1.2.0.11	JKI Package Network	JKI

Figure 3-2 The JKI VI package manager with the installed packages. Example "JKI REST Client" is necessary for read/write onto the telemetric values.

To install the LabView palette(drivers) for the ESP301 controllers, it is possible to download the full palette on the weblink below or directly from the folder New LCU drivers.

https://www.newport.com/medias/sys_master/images/images/hf8/h1e/8797091758110/ESP301-LABVIEW-Drivers.zip

Copy the folder Newport ESP301 under the folder user.lib indicated below :

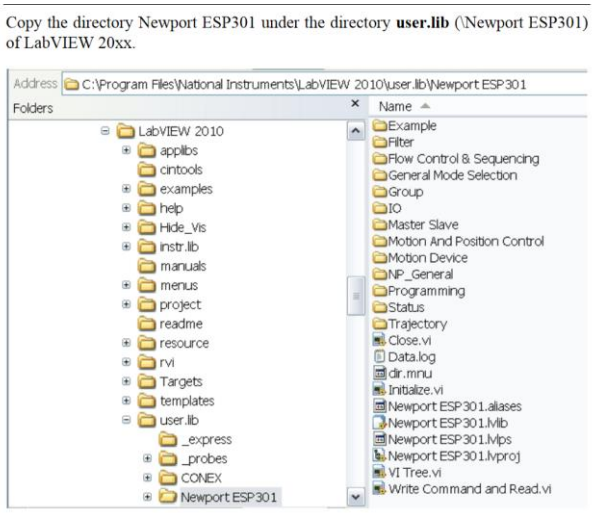


Figure 3-3 Link of installation of the Newport ESP301 LabView library

Once LabView restarted, it is possible to see the palette of functions of the controller.

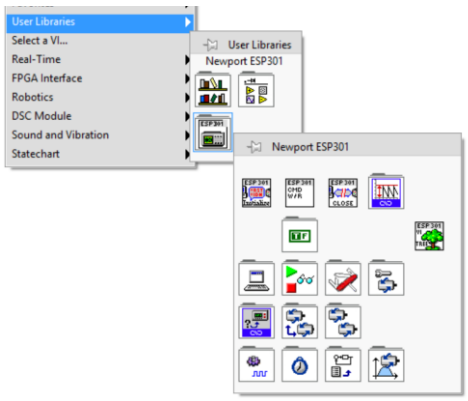


Figure 3-4 After installation of the LabView library, we can use the VI's under the user libraries

For the PCI ADLINK card, simply install the DAQ-LVIEW PnP V1.30 which will automatically install the LabView palette at the LabView user libraries as for the ESP controller.

4 CONCEPT OF THE NEW LCU

The LCU is conceptually done with 2 operating modes. 2 types of servers are built in one project.

The first type are used as webserver communications. The web resources communicate with the sequencer trough web URL. Each time the Sequencer send an URL where it can define some parameters values the corresponding VI executes ones and wait until the Sequencer call him again. Most of parameters are linked via global variables which make possible to use them everywhere between different VIs. All global variables can be found in the Global Variables folder.

The second type of servers are the hardware servers. Those VI's are standard VI's which are not linked with the web methods URLs. When the LCU start, those VI's are executed at start and stopped at the main VI stop. All hardware servers are running continuously even if nothing is asked from them. The communication with the hardware servers is made through the global variables we have defined above.

Finally, the FrontPanel.vi is the main VI where it refreshes all values and send commands to the hardware servers when required by the Sequencer of by the operator.

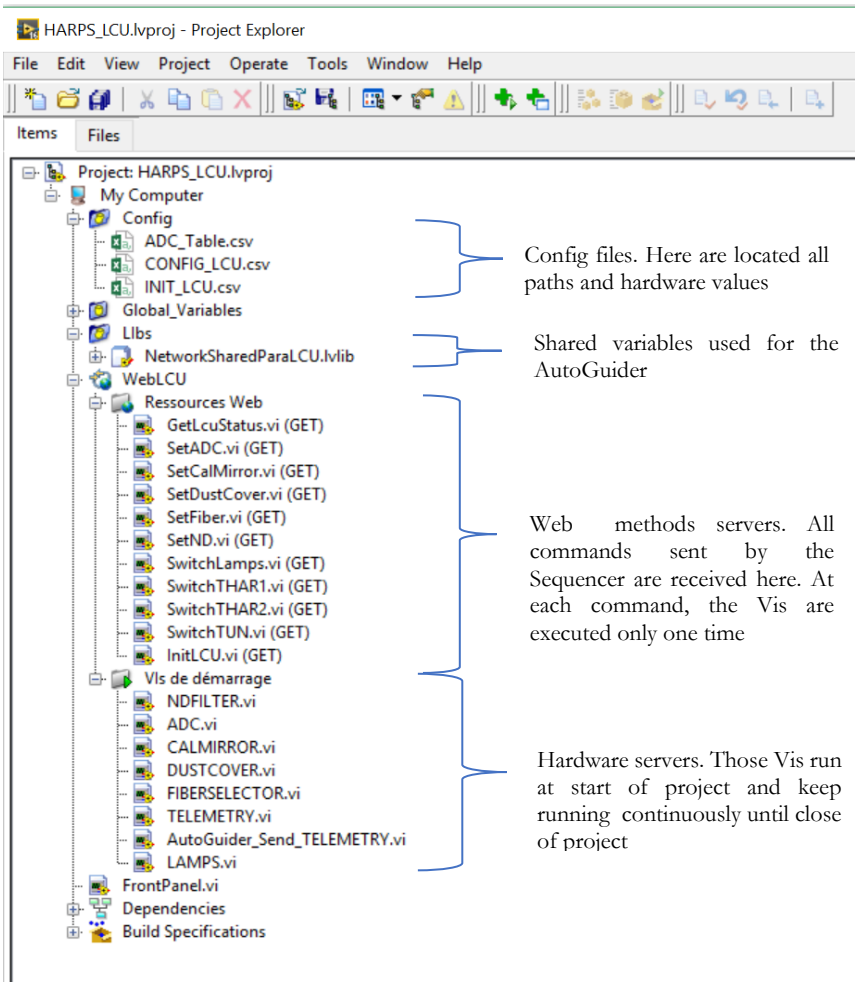


Figure 4-1 LCU project. The project is separated between two types of servers where the communication between them is made through global variables

All hardwares VI's are maid as the architecture shown below. A state machine is created where some initiate actions from the operator or from the Sequencer make pass the hardware from one state to the other one. Totally four states are described here :

OFF : The hardwares' VI is not started. The LCU is not started. This state is normally seen only by the Sequencer.

STANDBY : This is the first state once the LCU is started. In this state, the ESP301 controller can communicate with the LCU (ask of ID for ex.) but the motors are off. A movement with the hardware is so not possible. This allow the operator to work on the hardware safely.

ONLINE : When the initialization is started, the motors are turned on and the hardware search for the home position (kept as default values in the hardware config), then the hardware moved at its default position which is defined in the config file INIT_LCU.csv. Once the initialization done, the hardware is waiting for commands to move.

FAIL : This mode occurred only if some errors appears during an initiate action. If this is the case, the error will be treated and the hardware will go in standby mode where the motors would be turned off. An action of initialization is required to turn ON the motors and make a proper initialization.

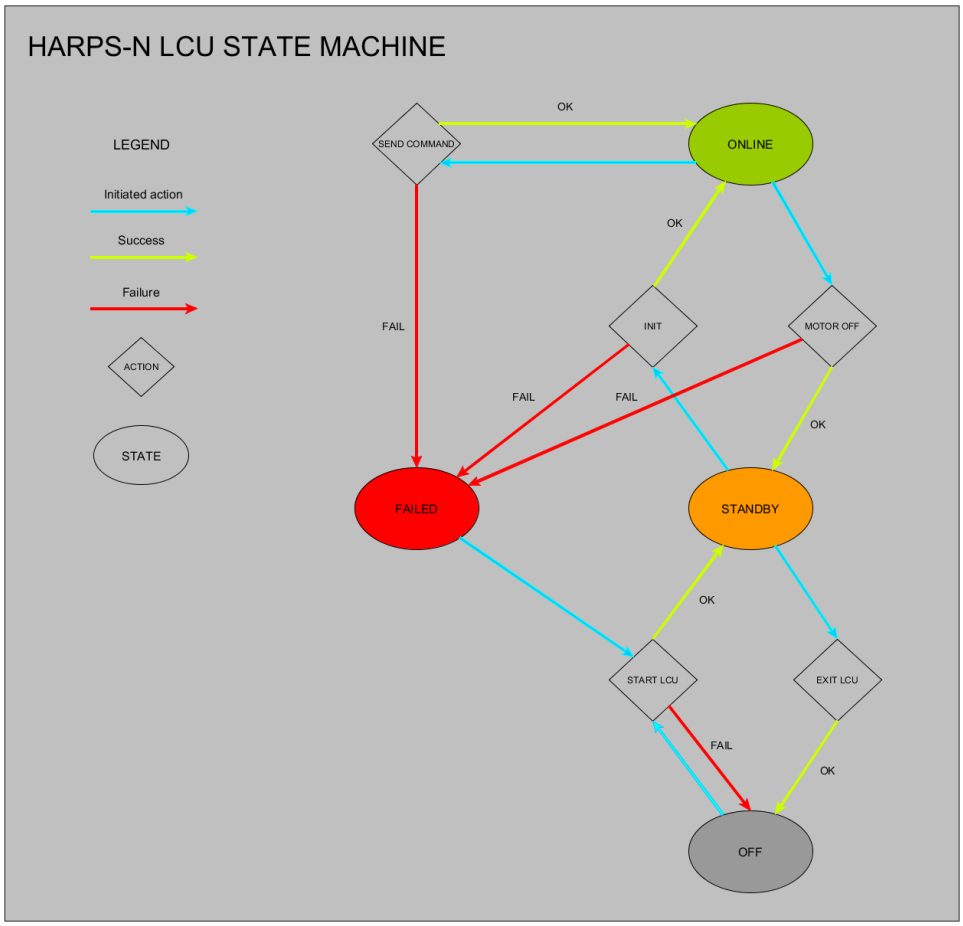


Figure 4-2 LCU State machine of all hardwares. All VI's are identical and so makes much easier to understand the architecture of coding

5 REMARKS AND NOTES

5.1 Launch of VI's from the web server

To be able to launch the LCU from a web server we need to create an executable application which will create the webserver.

We make an example with the hardware DUSTCOVER.

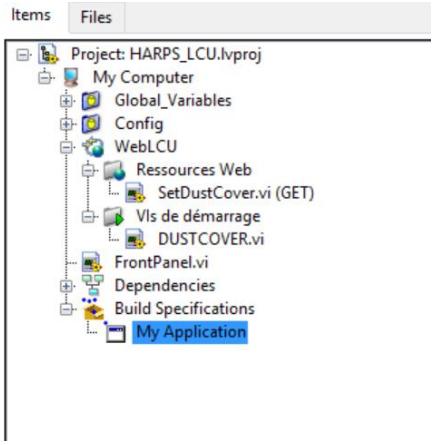


Figure 5-1 Example of project with only one hardware

In the properties of build of the application, it is important to set the VI's as below and select the following case (otherwise you can have directories errors).

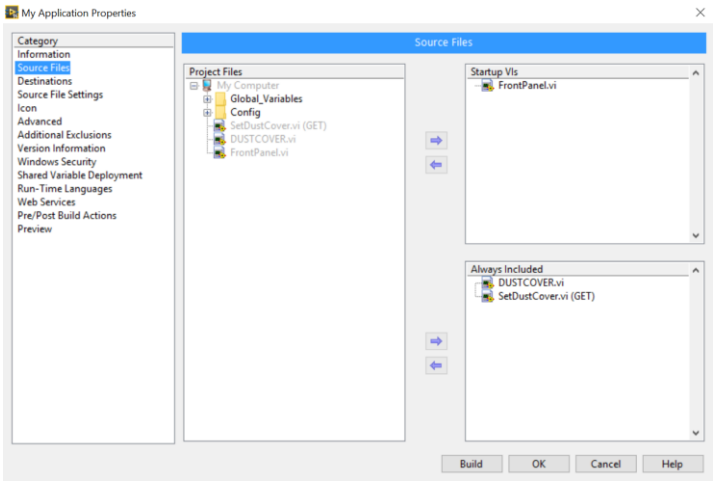
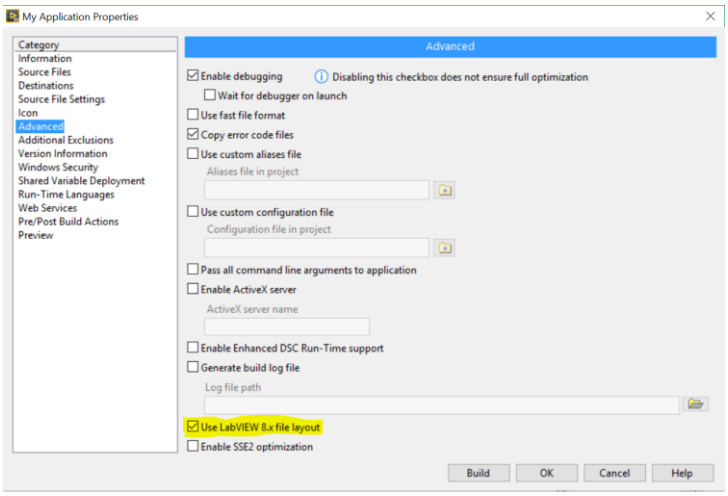
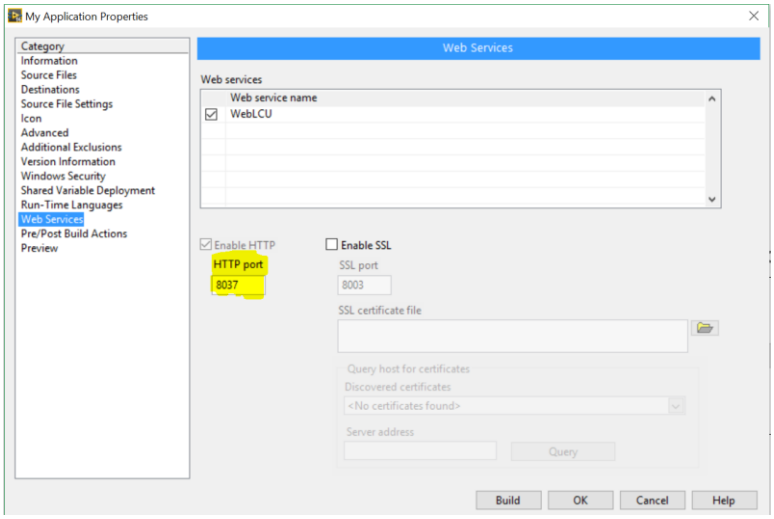


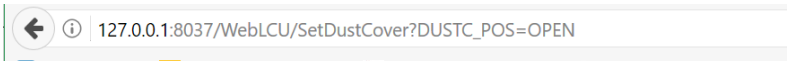
Figure 5-2 Only the FrontPanel.vi is inside the starting VIs, rest is in "always included" panel



In the web service the port number is defined and this will be the webserver port.

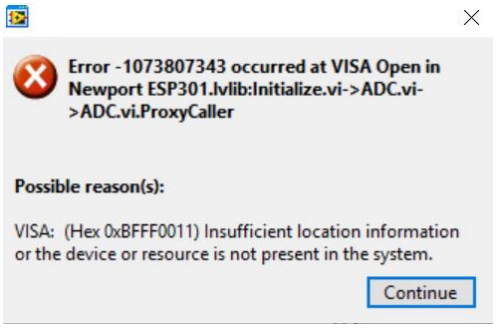


We can then run the application.exe and set the commands through the webserver by putting the parameters into the web url as below :

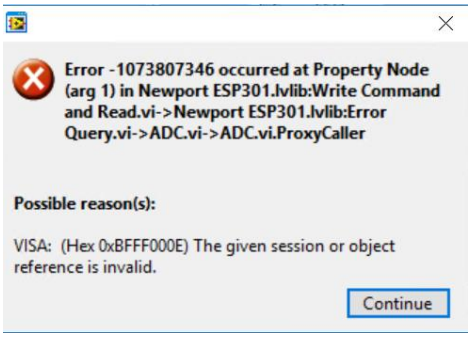


5.2 List of errors from the new LCU

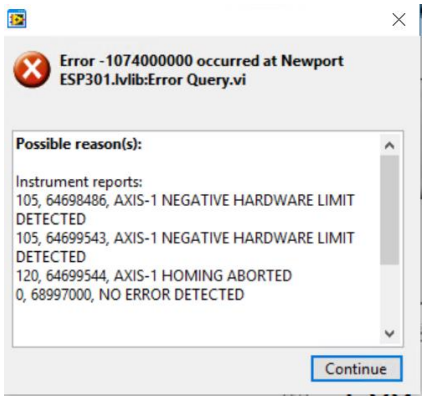
Error -1073807343 occurred at VISA Open in Newport ESP301.lvlib:Initialize.vi->ADC.vi->ADC.vi.ProxyCaller



Error -1073807346 occurred at Property Node (arg 1) in Newport ESP301.lvlib:Write Command and Read.vi->Newport ESP301.lvlib:Error Query.vi->ADC.vi->ADC.vi.ProxyCaller



Error -1074000000 occurred at Newport ESP301.lvlib:Error Query.vi



Those errors appears only with the controllers ESP301 (if usb are not connected, if some hardware reach the hardware limit or if we are asking a communication after closing it). This will be managed by the error handler.

5.3 Writing to Telemetry

The new LCU is writing to the telemtry with the following ulr :

http://ntcs-glassfish1.tng.iac.es:6080/ntcs-cacheservice/rest

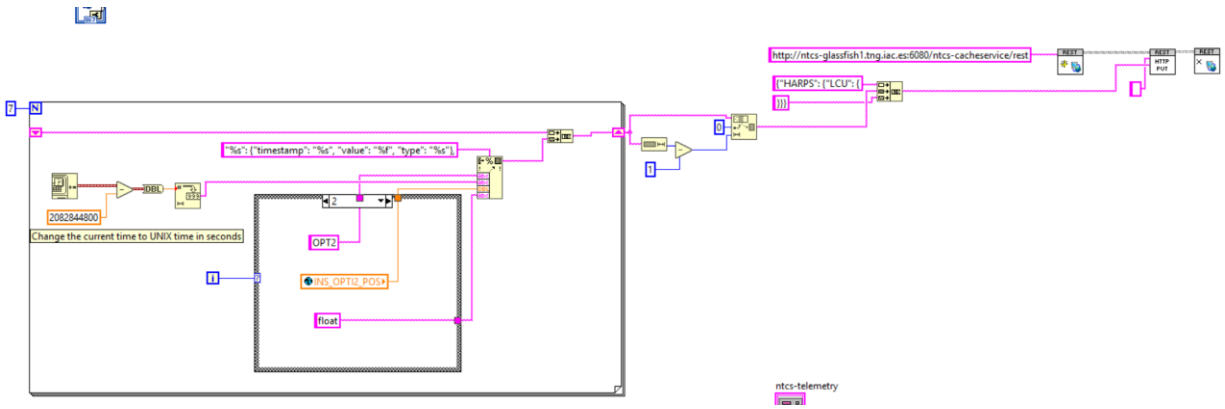


Figure 5-3 LabView code showing the construction of the body request to be sent to telemetry server. The rest architecture is used (put command is used to write values) in JSON format.

The body request is built in the manner where a for loop scan all parameters written into the telemetry and build in consequence the adequate body. The telemetry is written every 2 seconds. There is no label of hardware positions but only rough motor values. So that the operator can clearly see the exact and actual motor position instead of having a labeled position which potentially can be wrong. All motors positions for each hardware corresponding to a specified position are described in the CONFIG_LCU.csv in the config folder of the LabView project.

5.4 Communicating with the AutoGuider

The AutoGuider has been previously communicating with the old LCU through some shared variables.

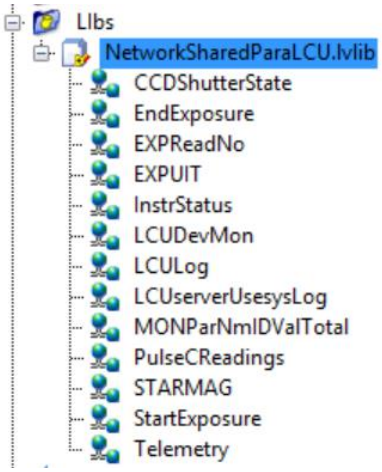


Figure 5-4 Shared variables folder from the new LCU. They are all located into the libs folder.

To be able to see what variables are used by the AutoGuider from the old LCU, the old LCU was started and autoguiding command sent with having a look in the shared variable device manager.

For the new LCU, the same method is used. The same shared variables are defined in the LabView project which are developed once the LCU launched. Potentially the AutoGuider do not use those variables but the software cannot work with undefined variables. So that the new LCU is providing fix values except for the telemetry.

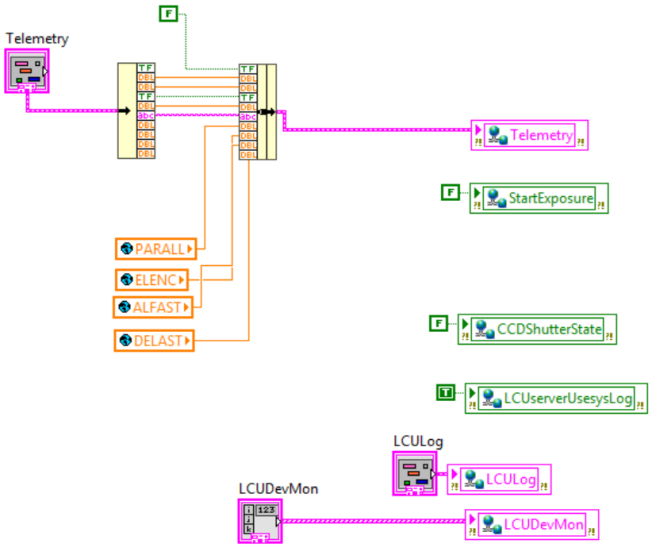


Figure 5-5 Fixed values written into the shared variables server of the new LCU which the AutoGuider reads

We can say actually that the shared variables are perfectly working and sending temeletric values to the AutoGuider.

Rest to be done to find all VI's where the old LCU IP address is defined. Here below are listed all VI's where the old IP is defined and now corrected with the new LCU one.

- AG_DevMon.vi (double click on Telemery variable. The link will appeared. It was hidden on the front panel…
- AG_Exposedo.vi
- AG_Acquiredo.vi

There is certainly more VI's where the IP address is defined. It is important to check that before deploying the project. Due to time limitations, the full project has not been verified.

After the change, it is important to deploy the distributionAG folder and all executables used to start the AutoGuider. That means the following executables have to be built :

1. distributionAG folder (takes 12 min to be correctly developed)
2. AG_DevMonitor
3. simpleAGClientEXE
4. simpleAGServerEXE
5. syslogCollector

The built directory has not to be changed from the original one. It will replace the old executables. In case of change, the address onto the shortcut of the AutoGuider has to be changed (shortcut proprieties).

6 NEW LCU MODIFICATIONS TO BE DONE

1. General Initialization button on LCU
2. Set Pos to update
3. Read the position on hardware during homing movement
4. Read position of hardware at STANDBY state
5. Create an installer for the new LCU (no more building projects)
6. Date/time on log
7. Save the log report
8. Add 2 polarimeters offset on the ADC into a config file
9. Config file for all hardware (COM ports and axis)
10. Create a server showing the status of the LCU
11. Pos of motors inside the LCU status
12. Error handlers
13. Annotation on all VI's
14. User manual
15. Installation manual
16. Finalizing of writing into telemetry

7 CONCLUSION AND FUTURE PLANS

The new LCU seems working correctly. Minors changes and bugs have to be done (plan is to finish everything until end of September).

A simple user manual will be written as well as the installation manual (which is mostly described in the chapter 3 of this report) with which anybody will be able to prepare a spare machine of the new LCU.

One day will be dedicated to transfer the project to Geneva's hands with final tests and potentially one night observation and one additional morning for the teaching on the software.

Conceptually, the AutoGuider should be completely separate from the LCU. The actual LCU design has been made so that we "just" solve the problem but without really making something clean. In addition of that, the computer is crashing several times and restarting during operations. Probably due to an hardware problem. Finally, the AutoGuider software is so complicated that it is just impossible to add or modify something into it. Patches are possible but just make things worse.

My suggestion would be to realize a new AutoGuider, with a new computer that we would provide. The AutoGuider could then be an autonomous application where the new LCU will be modified (and so simplified) to not communicate with the AutoGuider anymore.

The communication with the Sequencer would be identical as with the new LCU (web methods with REST architecture) and with a NI Vision builder VI's which makes possible to do easy vision applications with LabView. We have a small experience with this and think that it would be possible to realize it in this way.

Finally, a small test setup can be designed in house to test the new AutoGuider. So that we can ensure us of correct working of the software before coming for the integration on site.