List of functions to communicate with ALR32XX in python

Manufacturer: ELC 59 avenue des Romains 74000 ANNECY - FRANCE
Phone: +33 (0)4 50 57 30 46 Fax : +33 (0)4 50 57 45 19
Website : www.elc.fr email : commercial@elc.fr

This library provides functions to access the different parameters of the ALR32XX programmable power supplies.
If you wish to program your own functions, the communication protocol can be found in the manuals in Appendix A.

List of functions for initialisation:

Function in the ALR32XX class	**Description**	**Input parameters**

__init__(c_name=' ')
Example :
X=ALR32XX ('ALR3206T')
Answer :
Connexion O.K
Port=COM22
Nom=ALR3206T
	Selection of the product and configuration of the port to establish serial communication.
	The initialization contains a procedure for automatic connection to the power supply.
		c_name : device name
		● 'ALR3203'
		● 'ALR3220'
		● 'ALR3206D'
		● 'ALR3206T'

List_ports () **Example :** X.List_ports() Answer : COM3 - Intel(R) Active Management Technology - SOL (COM3) COM22 - Périphérique série USB (COM22) COM1 - SUNIX COM Port (COM1)	Displays the list of active ports on the computer.	
Choix_port() **Example :** X.Choix_port() Answer : 'COM22'	Manual connection procedure. Displays the list of active ports and asks to select the power supply port to connect to.	
Deconnexion() **Example :** X.Deconnexion() Answer : 'OK \r'	Stops serial communication and switches the power supply to "LOCAL" mode	
IDN () **Example :** X.IDN() Answer : 'ALR3206T VERSION 1.2 \r'	Returns the identity of the power supply (name + version)	

List of functions for voltage/current control :

Function in the ALR32XX class	**Description**	**Input parameters**
Ecrire_tension (**value**=0, **channel**=1) **Example :** X.Ecrire_tension(15.025) <u>Answer :</u> 'OK\r'	Write a voltage setpoint on the selected channel. The value in Volt (V) evolving according to the parameters of the power supply.	**value** : desired voltage, defaulted to 0V ● Integer or character string depending on the characteristics of the power supply **channel** : channel on which to change the voltage, defaulted to 1 ● Integer from 1 to 3 This parameter is optional on 1-channel power supplies.
Ecrire_courant (**value**=0, **channel**=1) **Example :** X.Ecrire_courant(5, 2) <u>Answer :</u> 'OK\r'	Write a current setpoint on the selected channel. The value in Amperes (A) evolving according to the parameters of the power supply.	**value** : desired current, defaulted to 0A ● Integer or character string depending on the characteristics of the power supply **channel** : channel on which to change the current, defaulted to 1 ● Integer from 1 to 2 This parameter is optional on 1-channel power supplies.
Mesure_tension (**channel**=1) **Example :** X.Mesure_tension(1) <u>Answer :</u> 19,793	Measure the voltage in Volt (V) on the selected channel.	**channel** : channel on which to measure the voltage, defaulted to 1 ● Integer from 1 to 2 This parameter is optional on 1-channel power supplies.
Mesure_courant (**channel**=1) **Example :** X.Mesure_courant(1) <u>Answer :</u> 1,050	Measure the current in Amperes (A) on the selected channel.	**channel** : channel on which to measure the current, defaulted to 1 ● Integer from 1 to 3 This parameter is optional on 1-channel power supplies.
Consigne_tension (**channel**=1) **Example :** X.Consigne_tension(1) <u>Answer :</u> 32,000	Read the voltage setpoint (V) on the selected channel.	**channel** : channel on which to read the voltage setpoint, by default at 1 ● Integer from 1 to 3 This parameter is optional on 1-channel power supplies.

Consigne_courant (**channel**=1) **Example :** X.Consigne_courant(1) Answer : 1,000	Read the current setpoint (A) on the selected channel.	**channel** : channel on which to read the current setpoint, by default at 1 ● Integer from 1 to 2 This parameter is optional on 1-channel power supplies.

List of auxiliary functions:

OUT(**mode**=’OFF‘, **channel**=1) **Example :** X.OUT(‘ON’, 1) Answer : 'OK \r'	Connect or not the output terminals.	**mode** : control parameter ● ‘ON’ or 1 ● ‘OFF’ or 0 **channel** : channel to operate on, by default at 1 ● Integer from 1 to 3 This parameter is optional on 1-channel power supplies.
MODE (**mode**=’NORMAL‘) **Example :** X.MODE(‘NORMAL’) Answer : 'OK \r'	Change the output coupling.	**mode** : coupling parameter ● ‘NORMAL’ ● ‘SERIE’ ● ‘PARALLELE’ ● ‘TRACKING’ This function is not available for 1-channel power supplies.
TRACK(**mode**=’ISOLE’) **Example :** X.TRACK(‘ISOLE) Answer : 'OK \r'	Activate the tracking mode. This function uses the previous one to couple the outputs.	**mode** : tracking parameter ● ‘ISOLE’ ● ‘COUPLE’ This function is not available for 1-channel power supplies.
OVP(**value**=0, **channel**=1) **Example :** X.OVP (1, 32.200) Answer : 'OK\r'	Write an OVP (Over Voltage Protection) value in Volt (V) on the selected channel. The OVP is the programmable voltage limit.	**value** : desired voltage limit, defaulted to 0V ● Integer or character string depending on the characteristics of the power supply **channel** : channel on which to change the voltage limit, defaulted to 1 ● Integer from 1 to 2 This parameter is optional on 1-channel power supplies.
OCP(**value**=0, **channel**=1) **Example :** X.OVP (1, 6.100) Answer : 'OK\r'	Write an OCP (Over Current Protection) value in Amperes (A) on the selected channel. The OCP is the programmable current limit.	**value** : desired current limit, defaulted to 0A ● Integer or character string depending on the characteristics of the power supply

		channel : channel on which to change the current limit, defaulted to 1 ● Integer from 1 to 2 This parameter is optional on 1-channel power supplies.
OVP_OCP(**parameter**='OVP', **channel**=1) **Example :** X.OVP_OCP(‘OVP’, 1) <u>Answer :</u> 32,110	Read the OVP and OCP values recorded on one of the channels of the device.	**parameter** : reading target ● ‘OVP’ ● ‘OCP’ **channel** : channel on which to read the parameter ● Integer from 1 to 3 This parameter is optional on 1-channel power supplies.
Read_state_ALR(**parameter**=’OUT’) **Example :** X.Read_state_ALR(‘REM’) <u>Answer :</u> 'OK 1\r'	Read the status of various power parameters.	**parameter** : reading target ● ‘REM’ ● ‘TRACK’ ● ‘MODE’
Remote(**mode**=’REMOTE’) **Example :** X.Remote(‘LOCAL’) <u>Answer :</u> 'OK \r'	Change the communication mode of the power supply.	**mode** : activate or not the serial communication ● ‘LOCAL’ ● ‘REMOTE’
STO(**memory_box**) **Example :** X.STO(‘1’) <u>Answer :</u> 'OK \r'	Save the power configuration in a memory box.	**memory_box** : location of the backup ● Integer from 1 to 15
RCL(**memory_box**) **Example :** X.RCL(‘1) <u>Answer :</u> 'OK \r'	Recalling the configuration of a memory box	**memory_box** : location of the backup ● Integer from 1 to 15