

ATLAS Stave Quality Assurance testing program ChillerCtrl.py

The program ChillerCtrl.py purpose is to control the coolant flow through the ATLAS stave. The process can be operated either by a user at the controlling computer entering commands to devices or by reading a file with predefined list of commands to perform. This program is a simple text entry program that runs in a terminal session.

The following block diagram shows the links of the program to the devices & files.

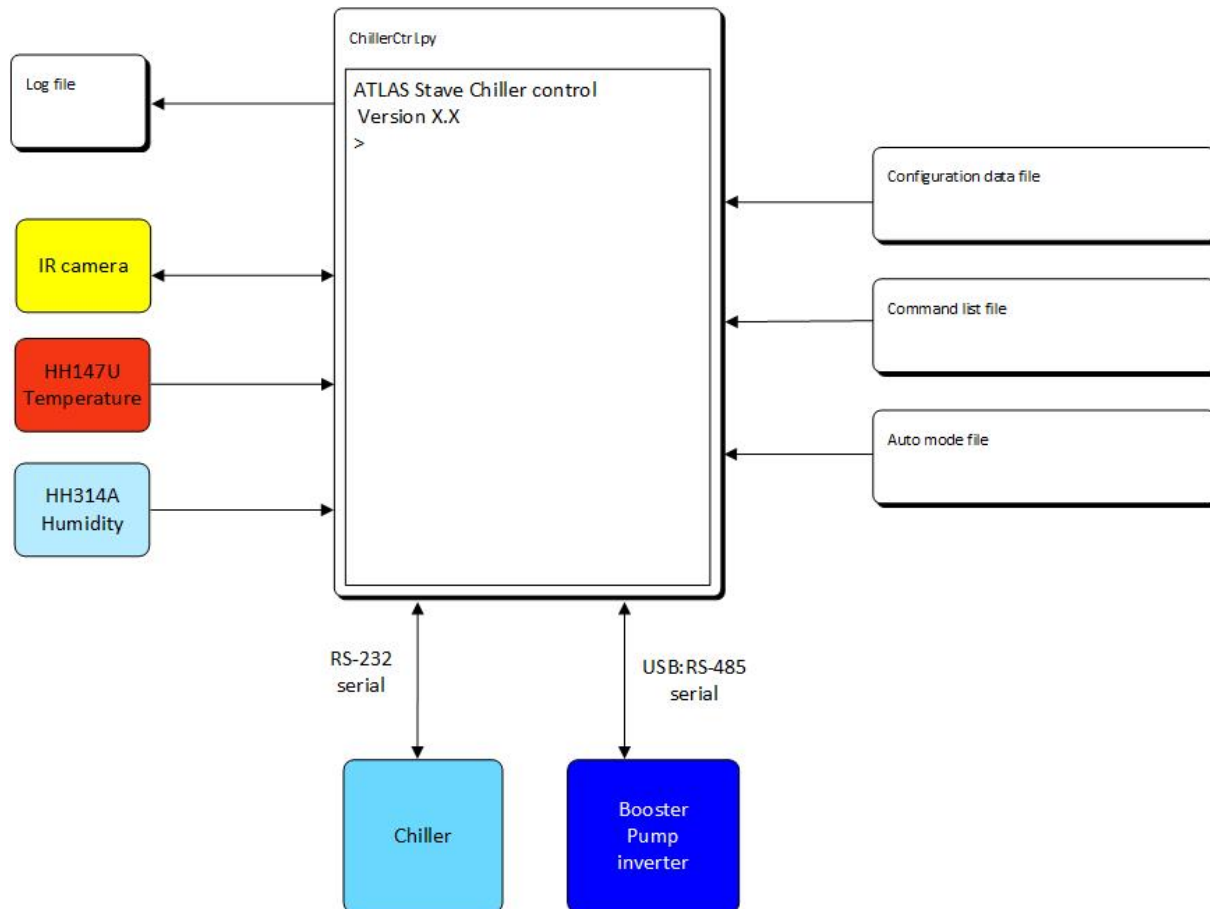


Figure 1. Block diagram of external files & equipment ChillerCtrl communicates with.

Program flow

Start the program in a terminal session.

The program will print to screen an identification statement (Stave coolant temperature & flow control) and the current version (V1.0)

First action is to create and open for writing a log file. Ask user for stave ID and locations of temperature sensors. (ex: T1 – enclosure temperature, T2 – chiller reserve temperature, ...) Write a header with information on date, version of code, stave ID, temperature sensor locations. Handle any issues with creating & writing to log file. This log file will be written to every N seconds with the time, temperature sensor(s) values, humidity? plus info on program actions (open & read files, etc.) and any command executed.

Next open & read the command list file. Inform user of the status of this action. If issues occur handle them. If no issues continue.

Next action is to open the configuration file and read data. Inform user the status of action. If issues occur handle them. If no issues, send a status command to all connected devices. If they respond continue. If not, inform the user and handle the issue.

In an infinite loop prompt the user for a command. When the internal command EXIT is entered, check the temperature of the coolant. If it is at room temperature (+/- 3C) write statement in log file, close log file and exit program. If coolant temperature is not at room temperature, send set point command to chiller for 23C. Inform user of this action.

Internal commands

There are commands internal to the program. Read, Loop, EndLoop, Wait, Exit, Question mark(?) and Abort.

The Read command requires a file name. This is the file that contains in a linear list the commands for all the devices connected to the controlling computer. The file will be opened and each command will be executed as each line is read in. Exception is the Loop & EndLoop commands.

The Loop & EndLoop define a looping construct. The Loop command requires an integer number of times the commands between Loop and EndLoop are to be executed. The commands between the Loop and EndLoop commands will be stored in the program and executed in order for the number of times dictated in the Loop command.

Wait requires an integer of the time the program is to pause before executing the next command in the list. (maybe have units too? S=seconds, M=minutes)

Exit is the normal procedure to break out of the infinite loop.

Question mark (?) will print on the screen all the commands plus the brief description that was read in from the command file.

Abort immediately sends the command to the chiller to set the temperature to 23C, ramp down the booster pump RPM to 0 and turn off the chiller pump.

The program is at all times waiting for a user entry.

The configuration file

The configuration file contains the information for setting the serial ports specifications. Baud rate, parity & stop bits..., which port is connected to what device. Commands that should be sent to devices in order to set them to a define start state. The file should have a header with date of creation and any notes of specifics for the defined configuration of ports & devices.

The command file

The command file lists all the command messages for all devices. The file should have a header with date of creation and notes. The commands will become part of the help section of the code, so a brief comment on the use of the command should be included with the command list.

The automated run file

The automated run file contains a linear list of commands to send to devices. The commands will be executed as read in. Exception is the Loop construct. Every command between Loop and EndLoop will be internalized and executed in order the number of times dictated by the Loop command. This should have a header with date of creation and description of purpose.

The external files format

All the external files will have a header with some information. The header and any comments will be preceded with the Python comment symbol "#".

1. The configuration file.

```
# ChillerCtrl.py configuration data. 20-Jul-2017
# This is the configuration information for communications with all devices.
# The serial ports (USB & RS-232) format is defined.
```

```
Command value # comment
```

```
Ex: Port 10 # USB port to booster pump inverter
    BAUD 1 # Set the baud rate to 1200
```

2. The command list file.

```
# ChillerCtrl.py command list 20-Jul-2017
```

```
User command : equipment command # Brief description
```

```
Ex: setpoint : SP # Sets the temperature of the coolant.
    temp: PT      # Query the current process temperature
    Poll : POLL   # A non-op used to test communications
```

Pumpstat : P402 # Returns inverter module status

3. The automated run file.

```
# ChillerCtrl.py automated run command list 20-Jul-2017
# This list sets the chiller to oscillate between -20C and 30C with a period of 30 minutes
# and flow rate of 1 L/minute (inverter set at 22RPM)
```

User command value # comment

```
Ex: setpoint 20   # Start run just under room temperature
   Wait 10m     # Give system time to reach equilibrium
   Loop 10
   Setpoint -20   # Start of oscillation temperature profile.
   Wait 20m
   Picture     # IR camera takes photo.
   Setpoint 40
   Wait 20m
   Picture     # IR camera takes photo.
   EndLoop
   Setpoint 23   # Return to room temperture
```

4. The log file. Example

```
# ChillerCtrl.py Version 1.0 log file
# Created 20-July-2017 10:35 am
# Stave ID = xyz
# T1 = enclosure, T2 = chiller, T3 = stave inlet, T4 = stave outlet
10:35:01 Open ChillerCmds.txt Success
10:35:10 Read ChillerCmds.txt Success
10:35:11 Opened ChillerConfig.txt success
10:35:20 Read ChillerConfig.txt Success
10:35:23 POLL Success
10:35:25 P402 3 Success
10:35:27 tempOn Success
10:35:29 hunidityOn Success
10:35:30 cameraOn Success
10:35:31 SP -10
10:35:32 wait 10m
10:35:34 23.0 15.0 23.2 22.8 1%
10:35:36 23.1 5.0 20.2 21.7 1%
.
.
.
12:55:00 TakePhoto
.
```

.

13:25:30 SP 23

13:25:31 Wait 10m

13:35:30 Stop

Program quit at 20-Jul-2017 1:35 pm