

- Algorithm for Master_Main

1. Start
2. Create the thread for CHEMIN_Server and Run
3. Create the thread for Power_Server and Run
4. Create the thread for Telecom_Server and Run
5. End

- Algorithm for CHEMIN_Server

1. Start
2. Wait for request from Client
3. If request arrives, read the message from that request
4. Wait for 7 seconds
5. If thread is interrupted, set the message to power off
6. If message is CHEMIN_ON, set the CCU to True
7. Create the CHEMIN_Client and make the client listen to port no. 9013
8. Get the power thread from RoverThreadHandler and start that thread
9. If message is power on, create a new variable called process and initialize it to false
10. If CCUYes is true, create the new CHEMIN and call its function CHEMIN_Process and catch the value returned by that function into the variable named CHEMIN_Process.
11. If CCU is not true, log the message that CHEMIN don't have permission from CCU yet.
12. If process is true, create the new CHEMIN_Client and make the client to listen on port no. 9002
13. Get the thread from RoverThreadHandler for that Client and start that thread
14. If message is power off, call the CHEMIN_POWER_OFF function of CHEMIN module.
15. Stop getting request from CHEMIN_Client and close all the resources.
16. End

- Algorithm for CHEMIN_Client

1. Start
2. Call the getRoverSocket function and call the getRoverSocket method on the object returned by the previous function
3. Call the getPort method on the object returned by getSocket method.
4. Check if the value returned by the getPort method = 9013 then create the output string
5. Read the power requirement from GlobalReader and convert that into json object.
6. Send that json object by output string to CHEMIN_Server
7. If the value returned by getPort function = 9002, read the image from GlobalReader and convert that into json object.
8. Send that json object to CHEMIN_Server
9. End

- Algorithm for CHEMIN_Process

1. Start
2. Get the current thread and make the current thread to sleep for 2 second
3. If the FunnelContamination is removed, start the Cryo Cooler and wait for 5 second
4. Again wait for 2 second
5. Check If CHIMRA has any sample or not
6. If there is any CHIMRA sample then wait for 2 second
7. If SampleCellContamination is removed then wait for 2 second
8. Check if FunnelPiezo is ON then PlayMusic and wait for 5 seconds
9. Start the SampleCellPiezo to ON and Start PlayMusic
10. Start the Xray and wait for 10 seconds
11. Read CCD
12. Try to create the XRD json file and if created, do nothing
13. Switch off the FunnelPiezo
Switch off the SampleCellPiezo
Switch off the XRay
14. If any of condition above is not satisfied then print appropriate message in returned false.
15. End

- Algorithm for Power_Server

1. Start
2. Start listening from the socket and read the message if any
3. Create the new POWER_Client and make the client to listen on port no. 9008
4. Get the thread for that client and start the thread
5. End

- Algorithm for Power_Client

1. Start
2. If the port is 9008, get the output string from the rover socket and send message power on to CHEMIN_Server
3. End

- Algorithm for Telecommunication_Server

1. Start
2. Start reading message and read the message if any.
3. Create the Telecommunication_Client and make the client to listen on port no. 9008
4. Get the thread from the RoverThreadHandler for that Client and start the thread.
5. End

- Algorithm for Telecommunication_Client

1. Start
2. If the port no. = 9008 then create the new output stream and pass the message "Telecommunication has received file successfully" to CHEMIN_Server
3. End

- Algorithm for Creating XRay Diffraction json file

1. Start
2. Create json from image
3. Create a new file called XRDiffracton.json
4. If file is not there, create a new file
5. Write the json object into that file named "XRDiffracton.json"
6. End

- Algorithm for PlayMusic

1. Start
2. Create the media player and play the sound which is in the file somebody.mp3
3. End