**ROXS Software User Guide:**

**Introduction:**

The ROXS software is a software package developed by the Chico State ROXS capstone team. The Package has been developed to reform x-ray spectroscopy a process preformed by the SSRL at SLAC.

**A Brief Overview:**

ROXS software can be broken down into several components, mainly the Spec Integration Webserver, the SLAP alignment software, and the gantry webserver. Communications and function calls were designed to be called through SPEC, a specialized command line-based software developed by Harvard for x-ray spectroscopy. These commands in spec call spec functions. These spec functions issue post curl-based commands to a Python based webserver. The Python webserver then calls functions for SLAP alignment written in python. On the existing ROXS test stand gantry communications are sent from python using the requests module to a ESP8266 REST webserver taking in JSON strings. The gantry webserver may then run stepper motor movements, zeros, led functions, and photoresistor intensity readings.

**SPEC Integration Webserver**

To illustrate how this software works, explanations will be paired with the code running in the current system.

A command is first called in SPEC.

|  |  |
| --- | --- |
| 1 | SLAP\_ALL\_I\_mm(150,300,1,3) |

**Excerpt 1.** SPEC SLAP\_ALL\_I\_mm Command Call

SPEC encodes a JSON string as a curl post.

|  |  |
| --- | --- |
| 1 | {zi:150,zf:300,loops:1,tests:3} |

**Excerpt 2.** SLAP\_ALL\_I\_mm JSON Args

The JSON string is then sent to the Python webserver with the IP of the webserver and the extension of the function.

|  |  |
| --- | --- |
| 1 | http://192.168.0.103/SLAP\_ALL\_I\_mm |

**Excerpt 3.** Python Server IP with Function Extension

The webserver has routes for each function. On receiving a JSON string, the elements are checked for too few or too many arguments. Then on a pass the function may then be called from python.

This process may be observed in the following code. First we see the class SLAP\_ALL\_I\_mmCommand.

|  |  |
| --- | --- |
| 1 | class SLAP\_ALL\_I\_mmCommand(systemCommand): |
| 2 | def \_\_init\_\_(self, app): |
| 3 | args = ['zi','zf', 'loops', 'tests'] |
| 4 | super().\_\_init\_\_('SLAP\_ALL\_I\_mm', \*args |
| 5 | app.add\_route('/SLAP\_ALL\_I\_mm', self) |

**Excerpt 4.** on\_post System Command Class Function.

The class for this function is written for the extension /SLAP\_ALL\_I\_mm. On a call it inherits the arguments from the systemCommand class. A class function of system command, on\_post may be observed below.

|  |  |
| --- | --- |
| 1 | def on\_post(self, req, resp): |
| 2 | kwargs = req.media |
| 3 | missing = set(self.args) - set(kwargs.keys()) |
| 4 | extra = set(kwargs.keys()) - set(self.args) |
| 5 | <<Check for bad JSON>> |
| 6 | else: |
| 7 | print(f"{self.commandString+'API'}, Robot {id}, {kwargs}") |
| 8 | resp.status = falcon.HTTP\_200 |
| 9 |  |
| 10 | func = getattr(Robot, self.commandString + "\_API") |
| 11 | func(\*\*kwargs) |
| 12 | resp.text = json.dumps({"success": True}) |

**Excerpt 5.** on\_post System Command Class Function.

The function to be called given by super().\_\_init\_\_('SLAP\_ALL\_I\_mm', \*args in combination with the line func = getattr(Robot, self.commandString + "\_API")

Creates the syntax for the function defined by func as ‘SLAP\_ALL\_I\_mm’ + ‘\_API’. The function then can be called by the line func(\*\*kwargs) passing a dictionary of parameters to the function SLAP\_ALL\_I\_mm\_API. This dictionary of parameters is the JSON string created with curl in spec. In the ROXS import in python the class function SLAP\_ALL\_I\_mm\_API may be found. Its definition is as follows.

|  |  |
| --- | --- |
| 1 | @classmethod |
| 2 | def SLAP\_ALL\_I\_mm\_API(cls, \*\*kwargs): |
| 3 | runtime,cls.ErrorData=cls.SLAP\_ALL\_I(float(kwargs['zi']),float(kwargs['zf']), int(kwargs['loops']), int(kwargs['tests'])) |

**Excerpt 6.** on\_post System Command Class Function.

The API command calls the SLAP\_ALL\_I\_mm command and passes the arguments as defined by the dictionary created with the JSON string. The output of the SLAP\_ALL\_I\_mm, an error array, is then stored in the python webserver as a class variable to be called later by other functions. This value is not reported back through SPEC as minimal information was designed to be passed through to SPEC for the simplest possible operator experience.

Standard operation:

When a crystal is installed or on a webserver startup a n error mapping command must be called. This may be called through either a SLAP\_ALL\_I\_mm where parameters are passed as mm or through SLAP\_ALL\_I where parameters are passed as energy levels. A Rowland scan may then be called either through distance arguments given in mm or in energy level as RunRow and RunRow\_mm respectively.

Optionally the ROXS system may be sent to error corrected positions through motor commands in terms of gantry position.

Trouble Shooting