

BUPT-1 Experiment Docking Template

1. General Description of the Experiment

This section should outline the overall concept of the experiment, detailing the tasks to be accomplished, components involved, and the ultimate goal and expected outcomes of the experiment, to provide a clear understanding for the installer.

2. Program Environment Dependencies and Installation Instructions

Describe all installation steps for the program on the default environment (Raspberry Pi+Ubuntu Server 20.04 / Atlas 200+Ubuntu Server 18.04), including specific version numbers for dependencies like Python. The installer should be able to complete the installation using a freshly imaged system following these steps.

3. Telemetry Requirements (Optional)

Satellite telemetry involves collecting data from a satellite's instruments and systems, then transmitting that data back to Earth for monitoring and analysis. Set predefined fields (with a total length of no more than 4 bytes for a single experiment) that can be transmitted in real-time to the ground station. These fields help determine if all necessary conditions for program execution are met, allowing ground control to send commands to the satellite if so.

Example:

Field Type	Field Length/bit	Description
Service Run Count	8	Increment by 1 each time the program runs, starting at 0

Input File Existence	2	Checks if the input file is in the specified folder. Default is 0, set to 1 if a file is present, 2 if the file input fails
Output File Existence	2	Checks if the output file is in the specified folder. Default is 0, set to 1 when an output file is present, 2 if the file output fails

4. File Description

Detail the program's file structure, including input and output files, excluding the environment. Use regular expressions for non-specific file names. If the file generation time or triggering action is unique, describe it separately.

Example:

- Integrated experiment between satellite and ground:
 - Input Folder: `/home/ubuntu/SatSipComm/Files/uploadJson`
 - Output Folder: `/home/ubuntu/SatSipComm/Files/downloadJson`
 - Protocol Processing: Code located in `/home/ubuntu/SatSipComm/`
 - Execute Protocol Processing Program: `/home/ubuntu/SatSipComm/handler.py`
 - Configuration info like port number, IP, etc., stored in: `/home/ubuntu/SatSipComm/config.py`

5. Experiment Procedure

Detail each step of the experiment, including specific commands to be executed and their effects. Satellite commands should be executed via the SSH protocol, typically using bash. Scripted commands should assume `sudo` doesn't require a password. Due to satellite transmission limits, individual commands should not be

too long; it's recommended to manage each experiment with a unified entry script.

Example:

- Create virtual network environment:

```
bash /home/ubuntu/source/setup-env.sh
```

- Start Program 1:

```
bash /home/ubuntu/source/start.sh
```

- Start Programs 2 and 3:

```
bash /home/ubuntu/SatSipComm/ranName.sh
```

- After waiting 30 seconds, execute a cleanup script to close all programs and clear the virtual network environment:

```
bash /home/ubuntu/source/remove-env.sh
```

- Package the output into a backup folder:

```
bash /home/ubuntu/SatSipComm/progName.sh
```