

KBSW210715 ROS - sync_get_stcm

sync_get_stcm, sync_get_stcmstcm

- -
 -
 -
 -
 -
-

- - Ubuntu 16.04 X86
 - ROS Kinetic
-

- Slamware Slamware
 - Apollo/Ares/Athena
 - Mapper
-

ROS-

1.Slamware ROS SDKROS

2.ROS sdksrccatkin_wsROS slamware_ros_samplesrccatkin

```
cd catkin_ws/src
catkin_init_workspace
```

3.

```
cd ..
catkin_make
```

4.

```
source devel/setup.bash
```

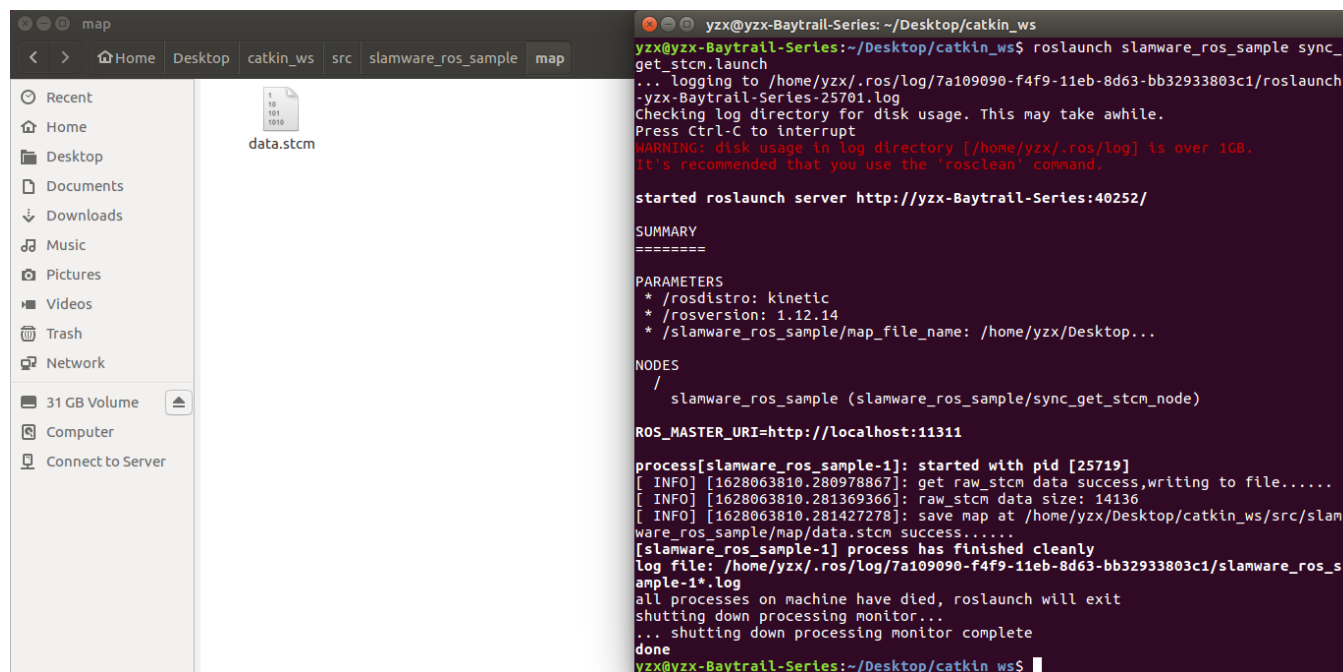
5.slamware_ros_sdk_server_node

```
roslaunch slamware_ros_sdk slamware_ros_sdk_server_node.launch ip_address:=10.6.128.141
//APip_address192.168.11.1
```

6.4sync_get_stcm_node

```
roslaunch slamware_ros_sample sync_get_stcm.launch
//launchmap_file_name
```

7.slamware_ros_samplemapdata.map



The image shows a file manager window on the left and a terminal window on the right. The file manager is displaying the contents of the 'map' directory, which includes a file named 'data.stcm'. The terminal window shows the execution of the 'roslaunch' command to launch the 'slamware_ros_sample' package. The output indicates that the launch was successful, with the 'sync_get_stcm' node starting and saving the map data to the 'data.stcm' file. The terminal also shows a warning about disk usage in the log directory and a summary of the launch parameters.

```
yzx@yzx-Baytrail-Series: ~/Desktop/catkin_ws
yzx@yzx-Baytrail-Series:~/Desktop/catkin_ws$ roslaunch slamware_ros_sample sync_get_stcm.launch
... logging to /home/yzx/.ros/log/7a109090-f4f9-11eb-8d63-bb32933803c1/roslaunch-
-yzx-Baytrail-Series-25701.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
WARNING: disk usage in log directory [/home/yzx/.ros/log] is over 1GB.
It's recommended that you use the 'rosclean' command.

started roslaunch server http://yzx-Baytrail-Series:40252/

SUMMARY
=====

PARAMETERS
* /rostdistro: kinetic
* /rosversion: 1.12.14
* /slamware_ros_sample/map_file_name: /home/yzx/Desktop...

NODES
/
  slamware_ros_sample (slamware_ros_sample/sync_get_stcm_node)

ROS_MASTER_URI=http://localhost:11311

process[slamware_ros_sample-1]: started with pid [25719]
[ INFO] [1628063810.280978867]: get raw_stcm data success, writing to file.....
[ INFO] [1628063810.281369366]: raw_stcm data size: 14136
[ INFO] [1628063810.281427278]: save map at /home/yzx/Desktop/catkin_ws/src/slam
ware_ros_sample/map/data.stcm success.....
[slamware_ros_sample-1] process has finished cleanly
log file: /home/yzx/.ros/log/7a109090-f4f9-11eb-8d63-bb32933803c1/slamware_ros_s
ample-1*.log
all processes on machine have died, roslaunch will exit
shutting down processing monitor...
... shutting down processing monitor complete
done
yzx@yzx-Baytrail-Series:~/Desktop/catkin_ws$
```

sync_get_stcm

```
int main (int argc, char **argv)
{
    std::string map_file_name;

    ros::init(argc, argv, "sync_set_stcm_node");
    ros::NodeHandle nh("~");
    ros::ServiceClient client = nh.serviceClient<slamware_ros_sdk::SyncGetStcm>("/slamware_ros_sdk_server_node/sync_get_stcm");
    slamware_ros_sdk::SyncGetStcm srv;

    //get file name from paramter server
    if (!nh.getParam("map_file_name",map_file_name))
    {
        ROS_ERROR("invalided file path.....");
        return 1;
    }

    //call service
    if (client.call(srv))
    {
        ROS_INFO("get raw_stcm data success,writing to file.....");

        //check whether the target folder exists. create one if not
        std::string dir;
        dir = map_file_name.substr(0,map_file_name.rfind("/"));
        if(access(dir.c_str(),0))
        {
            std::string cmd = "mkdir -p " + dir;
            system(cmd.c_str());
        }

        std::ofstream fout(map_file_name, std::ios::out | std::ios::binary);
        fout.write(reinterpret_cast<char *>(srv.response.raw_stcm.data()),(srv.response.raw_stcm.size()) * sizeof(srv.response.raw_stcm.front()));
        fout.close();

        ROS_INFO("raw_stcm data size: %zu",srv.response.raw_stcm.size());
        ROS_INFO("save map at %s success.....",map_file_name.c_str());
    }
    else
    {
        ROS_ERROR("Failed to get raw_stcm data.....");
        return 1;
    }

    return 0;
}
```