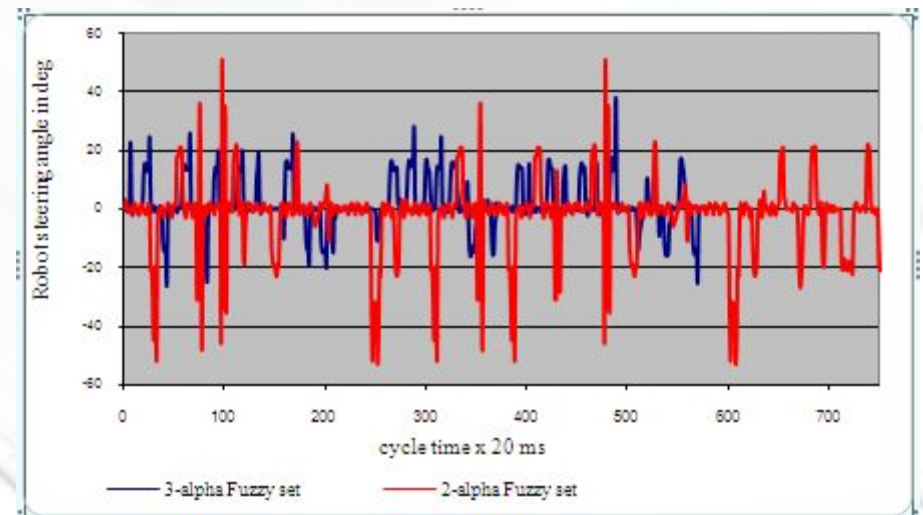
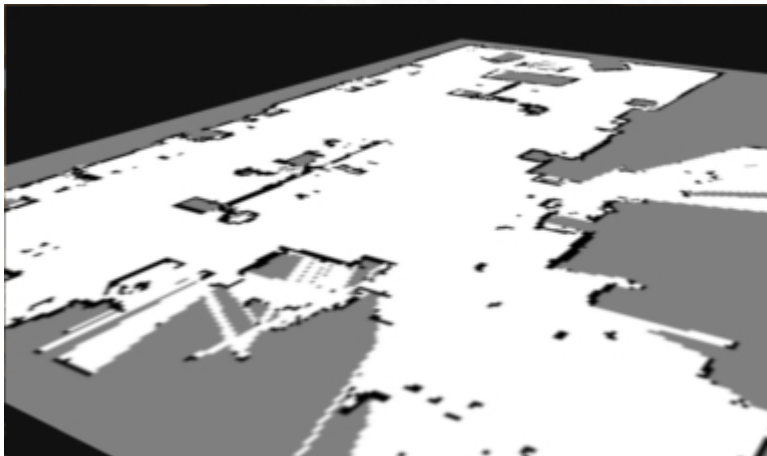


# ROSDASH



Web-based Visual-programming-based  
Configurable Dashboard Platform for ROS



# Why ROSDASH?

- Love robots, but have problems visualizing the data behind them?
- Use ROS, but tired of solving installation dependencies?
- Need dashboards specific for your research?

# Merits of ROSDASH

- Web-based: Just open browser and work!
- Visual programming: Drag and drop!
- ROS: Support topic, service, and parameter!
- Configurable: Change parameters and refresh the page!
- Platform: Log in and create your own dashboard!
- Extensible: Write your js code and link to it!

# How to Run?

- Roscore (talk to robots)
- Rosbridge (ROS => websocket)
- Roslibjs (websocket => js)
- ROSDASH (specify IP of rosbridge)
- Communication with robots works even off-line

# How to Edit?

- 3 pages for each dashboard
- Panel: Run widgets with ROS
- Editor: Edit widgets
- Diagram: Edit diagram representing connection between widgets and ROS topics

# Editor

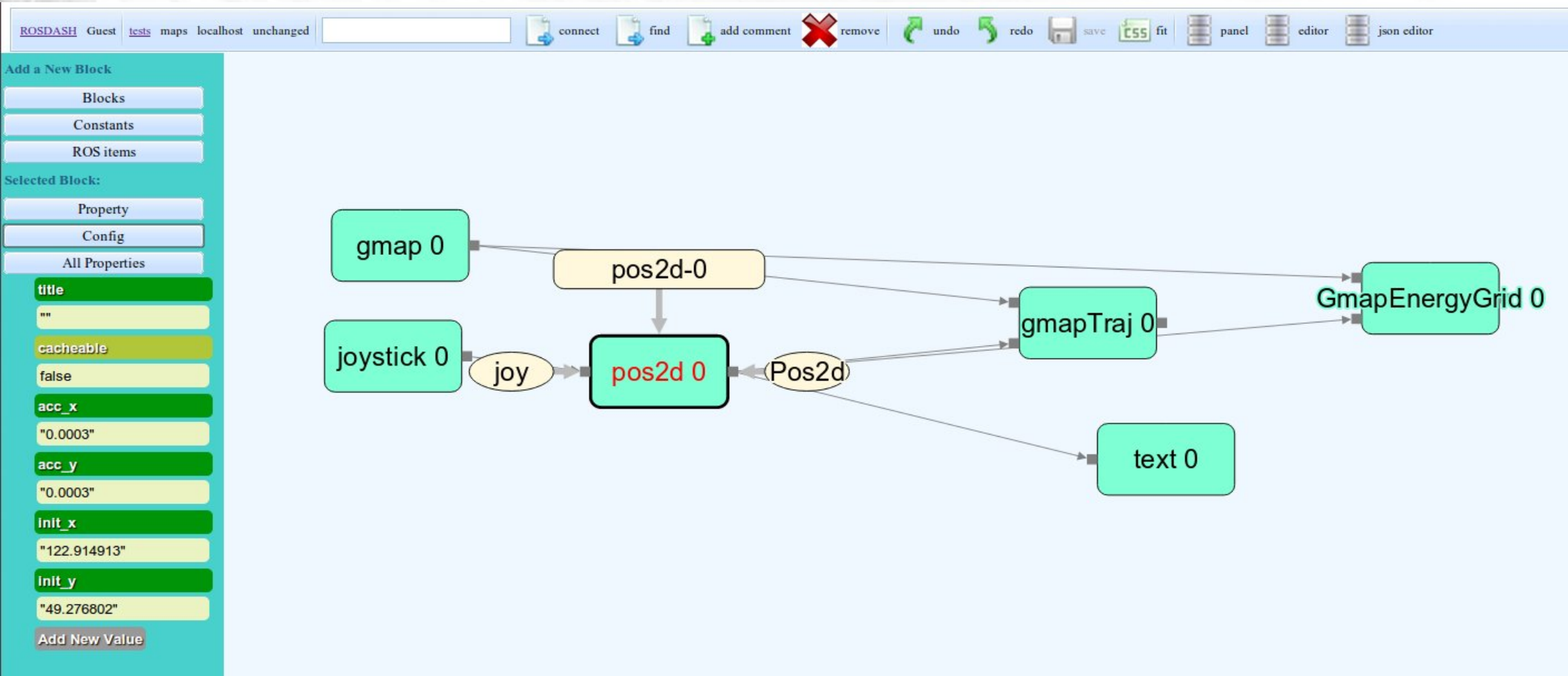
- Add, move, or remove widgets
- Change size of widgets

The screenshot displays the ROSDASH Editor interface. At the top is a toolbar with buttons for 'connect', 'find', 'undo', 'redo', 'save', 'panel', 'diagram', and 'json editor'. Below the toolbar is a status bar showing 'ROSDASH Guest' and 'tests maps localhost unchanged'. On the left side, there is a sidebar titled 'Add a New Widget' with a 'Widgets' button. Below this, under 'Selected Block:', are buttons for 'Property', 'Config', and 'All Properties'. A list of properties is shown with their current values: 'widgetTitle' is 'text 0', 'width' is '400', 'height' is '230', 'header\_height' is '16', and 'content\_height' is '180'. At the bottom of the sidebar is an 'Add New Value' button. The main workspace contains three widget containers: 'text 0' (highlighted in green), 'joystick 0', and 'gmap 0'. Each container has a title bar with a close button (X) and a maximize button (+).



# Diagram

- Flow network representing connection
- Add, move, or remove blocks
- Change config of blocks



# Dashboard

- Parse flow network
- All running widgets are independent
- Event emitter



# User-defined Widgets

- Json, js, css files
- Anywhere (local, internet, github)
- Include json file path in your config file

# A List of Widgets (1)

- Input: button, joystick
- Output: textbox, speech, table, chart

text 1

✕

+

```
{"header":{"seq":54,"stamp":
{"sec":1383958458,"nsec":109000000},"frame_id":""},"axes":
[0,0,0,0,0,0],"buttons":[0,0,0,0,0,0,0,0,0,0]}
```

joystick-0

unlock

✕

+




table 0

✕

+

Show 

25

 entries Search:

1

2

3

abc

def

Showing 1 to 1 of 1 entries 

◀

 Previous Next 

▶

speech 0

speak

✕

+

hahaha

toggleButton 0

✕

+

YES

text 0

✕

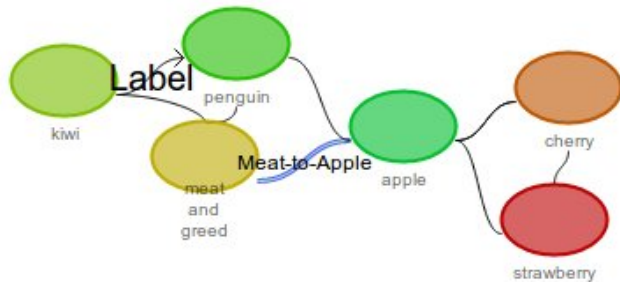
+

true

# A List of Widgets (2)

- Scientific: vu meter, network graph, Google maps, plot

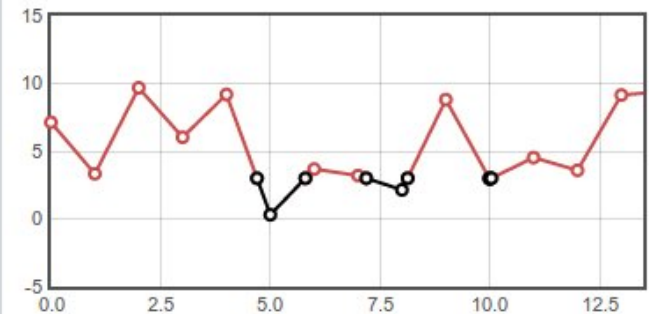
draculaNetwork 0



cyNetwork 0



flot 1



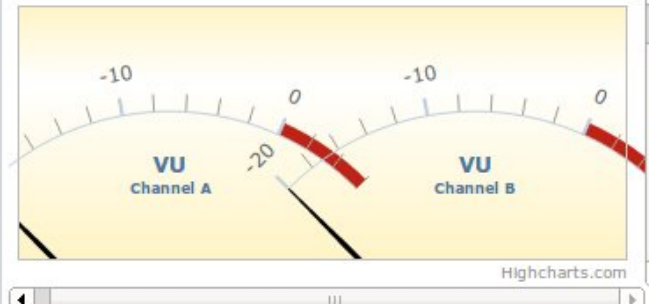
gmap 0 grids



Joystick-0 unlock



vumeter 0



# A List of Widgets (3)

- ROS: 2D model, 3D model, mjpeg, turtlesim

The screenshot displays the ROS GUI with several widgets arranged in a grid. The widgets are:

- table 0**: A table widget showing ROS topics, services, and parameters. It includes a search bar and a 'Show' dropdown set to 25 entries.
- joystick-0**: A joystick widget with an 'unlock' button and a red circular indicator.
- ros2d 0**: A 2D model widget showing a top-down view of a building layout.
- ros3d 0**: A 3D model widget showing a 3D perspective view of the same building layout.
- turtlesim 0**: A turtlesim widget showing a small robot (turtle) on a blue background.
- text 0**: A text widget showing a blank white area.

**table 0** details:

topics	services	params
/visualization_marker	/rosapi/delete_param	
/tf2_web_republisher	/static_map	
/cancel		
/rosout_agg	/map_server_1383958652880532739	
/rosout	/set_logger_level	
/map_metadata	/rosapi/has_param	
/map	/rosapi/get_loggers	
	/rosapi/search_param	

# A List of Widgets (4)

- Others: database, camera video, login with OpenID, json editor

# Future Work

- Rosdash.com
- What is needed in our lab?
- What is needed in others' labs?
- Website security
- Discussion in NCFRN?



# Steps to Create a Dashboard

1. Login with email account;
2. Go to your personal page;
3. Add a new page;
4. Open diagram page;
5. Pick some blocks to add;
6. Add some ROS topics from ROS item list, and connect them with widget blocks;
7. Open editor page;
8. Change the position of each widgets;
9. Run roscore and rosbridge;
10. Open panel page, and connect with rosbridge;
11. Run robots and see

# Steps to Create a Dashboard (1)

- 1. Login with email account;

Welcome to ROSDASH

## Welcome to ROSDASH !

A web-based platform of dashboards for roboticists and ROS users.

Sign in using your account with



Google



Yahoo!

Aol. AOL

[Social Login by Janrain](#)

table 0

Show 25 entries

Search:

user

user

user

[Fei Zhan](#)

[index](#)

[jake](#)

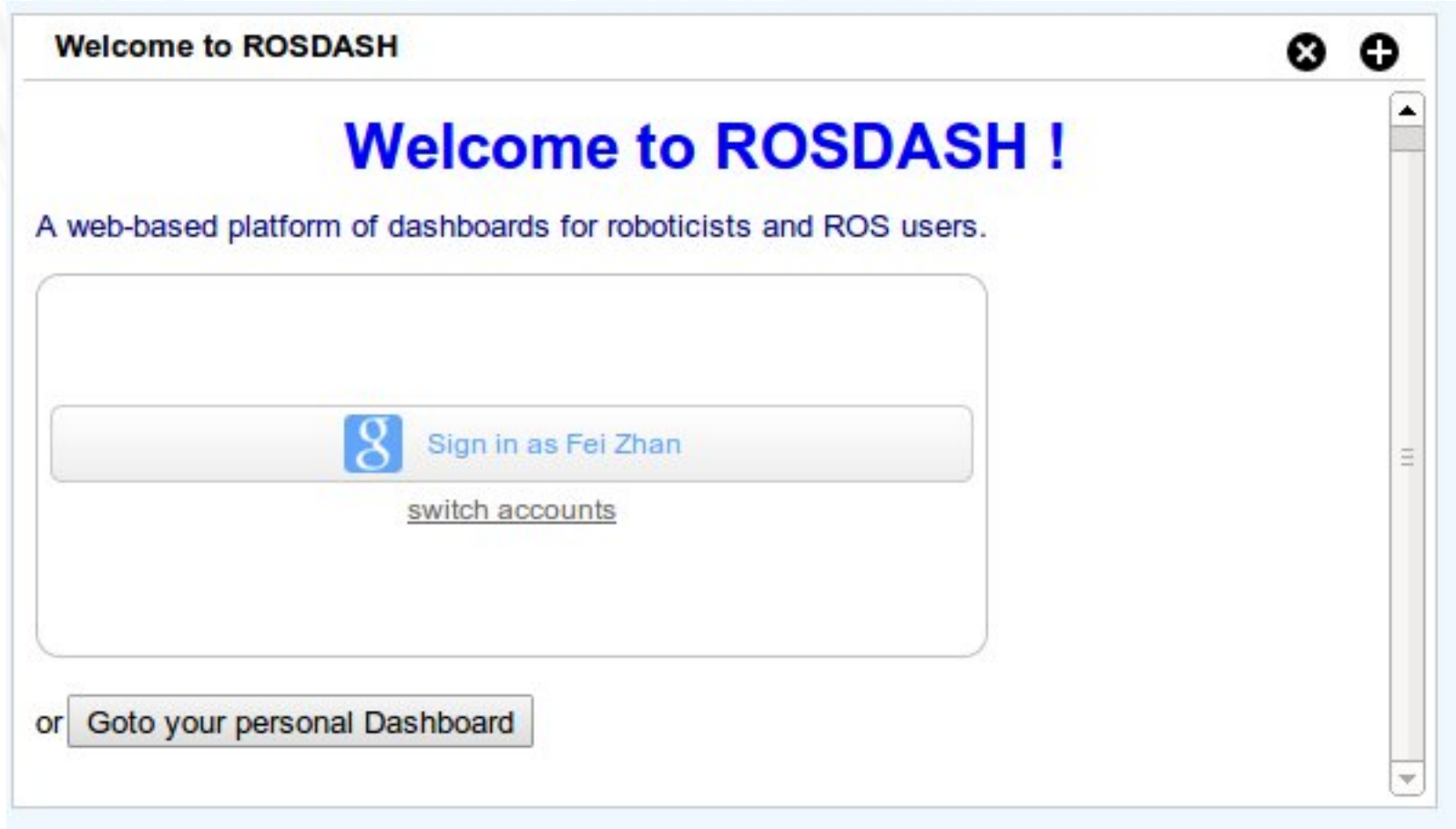
[tests](#)

Showing 1 to 2 of 2 entries

Previous Next

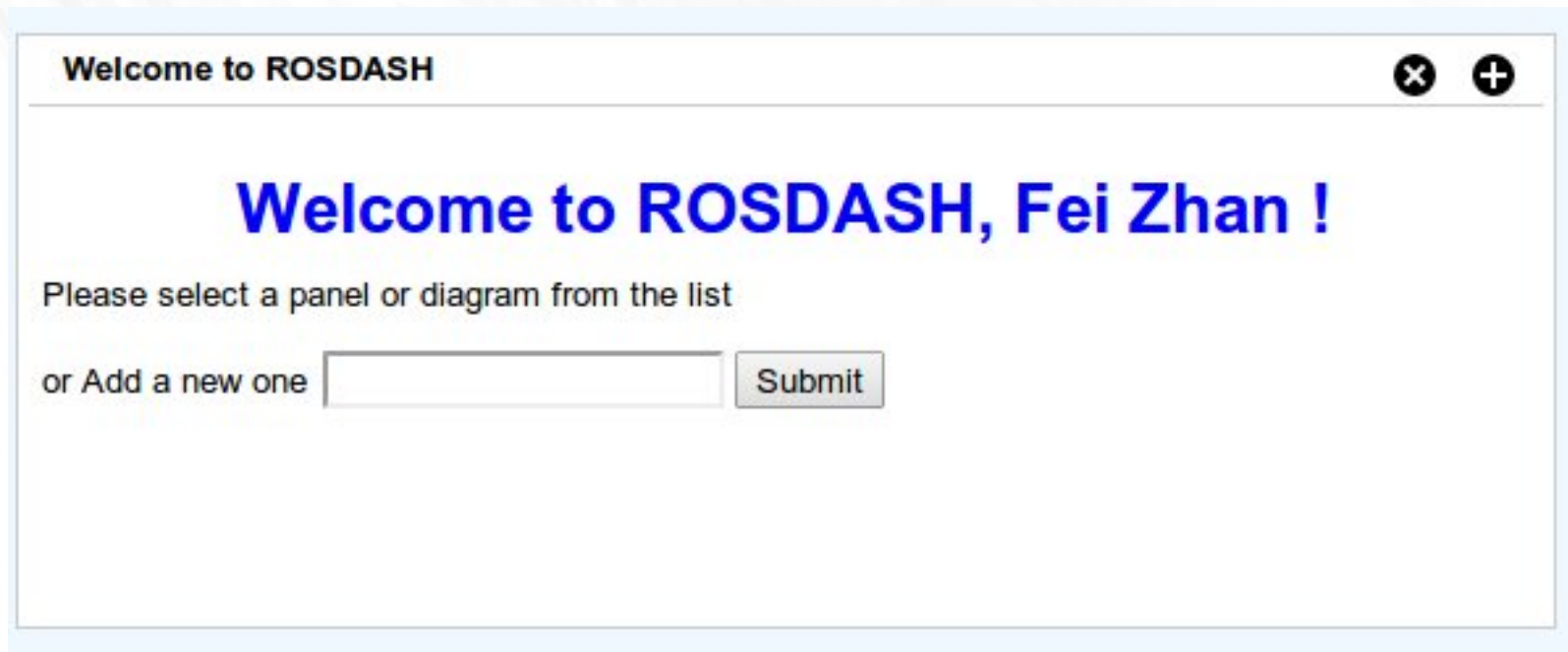
# Steps to Create a Dashboard (2)

- 2. Go to your personal page;



# Steps to Create a Dashboard (3)

- 3. Add a new page;



The screenshot shows a web application window titled "Welcome to ROSDASH". The window has a light blue border and standard window controls (close, maximize) in the top right corner. The main content area has a white background. At the top, it says "Welcome to ROSDASH" in black. Below that, a large blue heading reads "Welcome to ROSDASH, Fei Zhan !". Underneath the heading, a prompt says "Please select a panel or diagram from the list". At the bottom, there is a form with the text "or Add a new one" followed by a text input field and a "Submit" button.

Welcome to ROSDASH

**Welcome to ROSDASH, Fei Zhan !**

Please select a panel or diagram from the list

or Add a new one

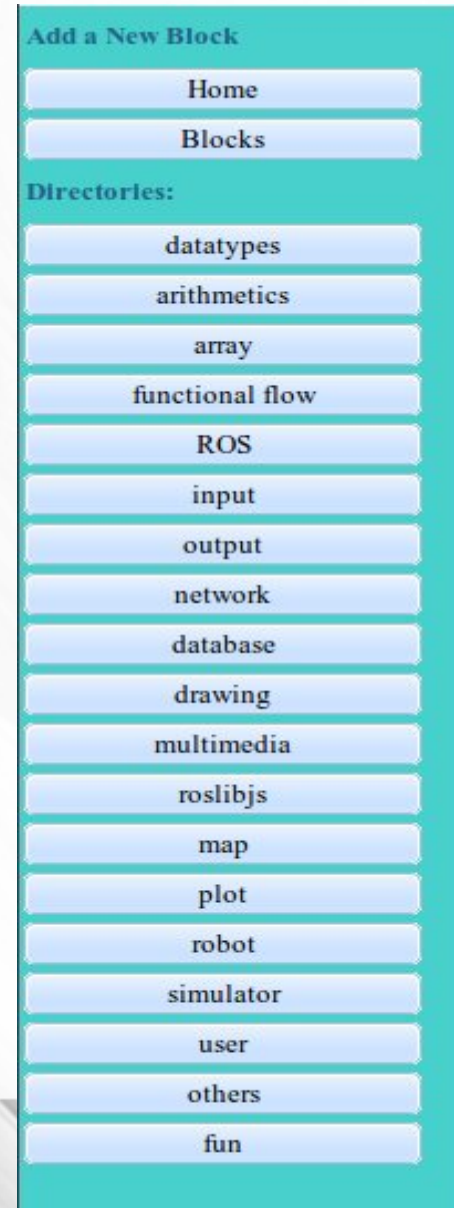
# Steps to Create a Dashboard (4)

- 4. Open diagram page;



# Steps to Create a Dashboard (5)

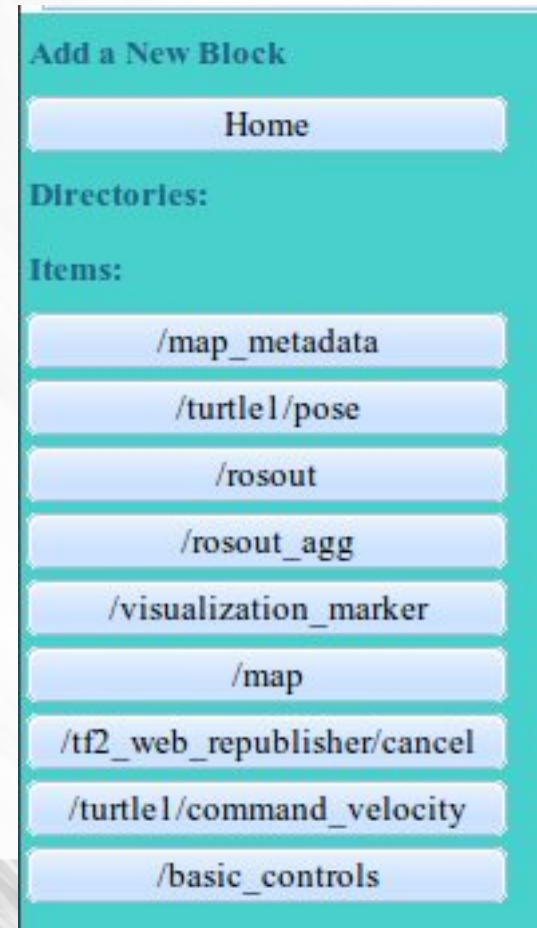
- 5. Pick some blocks to add;





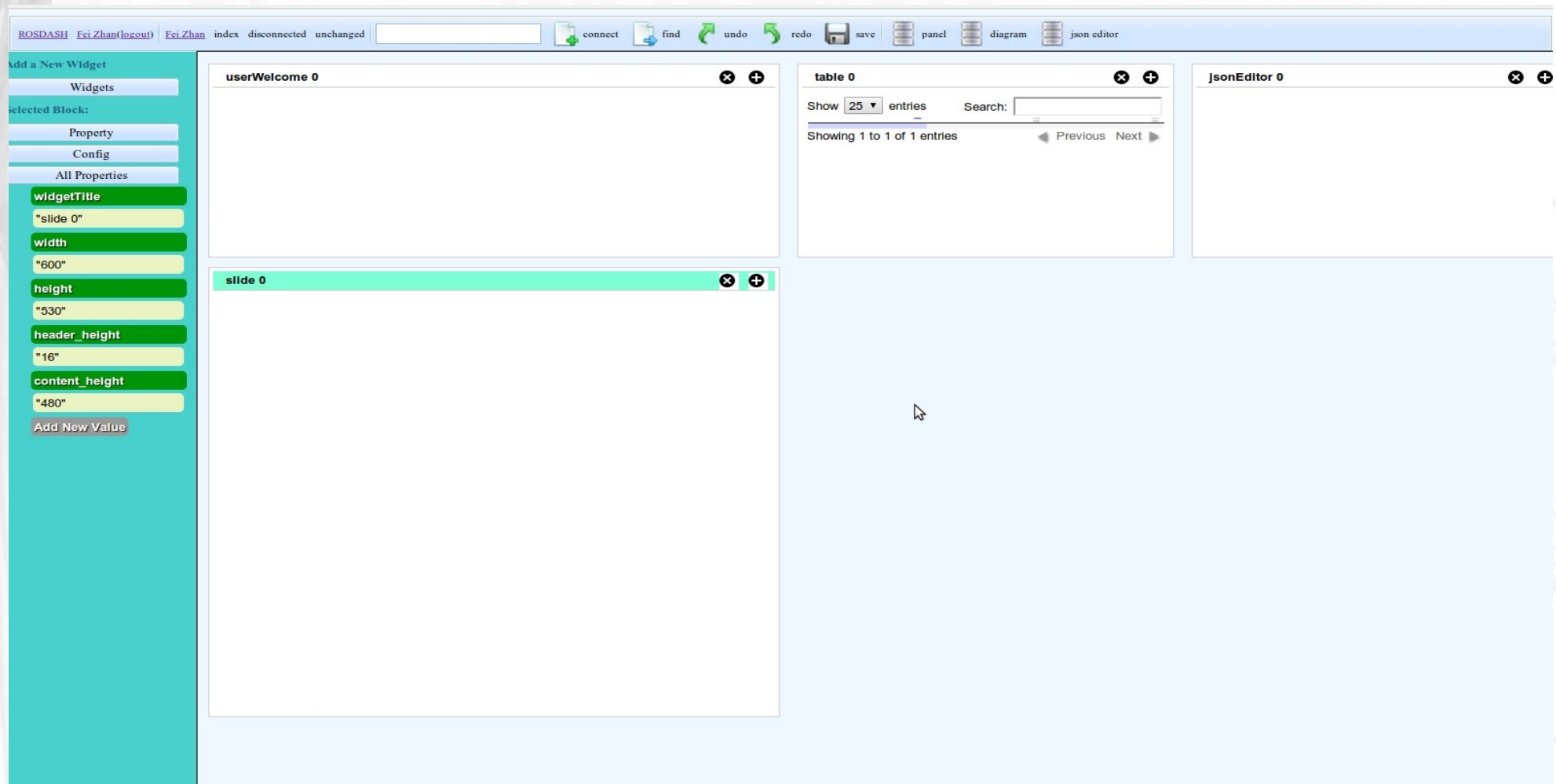
# Steps to Create a Dashboard (6)

- 6. Add some ROS topics from ROS item list, and connect them with widget blocks;



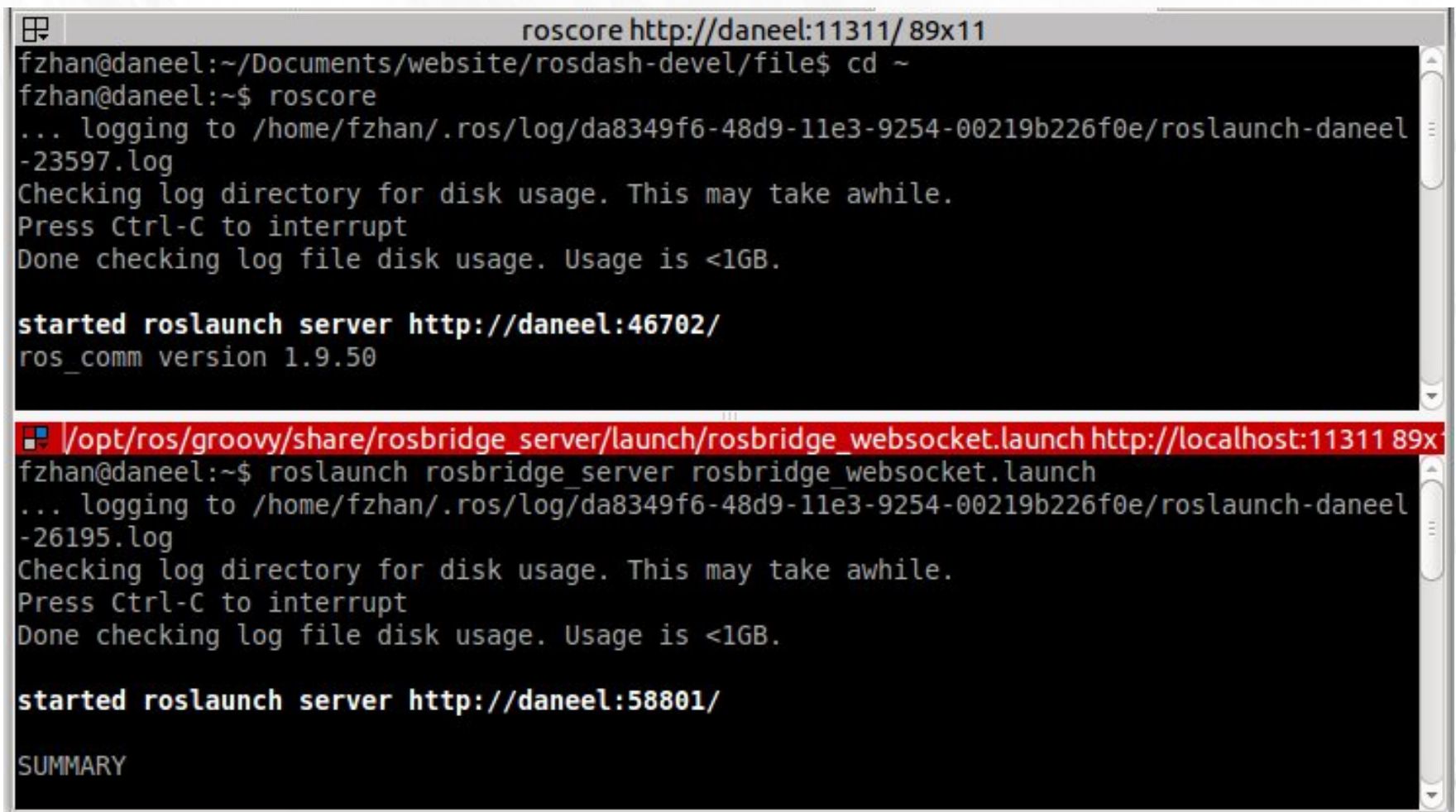
# Steps to Create a Dashboard (8)

- 8. Change the position of each widgets in the editor page;



# Steps to Create a Dashboard (9)

- 9. Run roscore and rosbridge;



The image shows two terminal windows. The top window is titled 'roscore http://daneel:11311/ 89x11'. It shows the user 'fzhan@daneel' navigating to a directory and running 'roscore'. The output indicates logging to a specific log file, checking disk usage, and starting the 'roslaunch server' on port 46702. The bottom window is titled with a red header: '/opt/ros/groovy/share/rosbridge\_server/launch/rosbridge\_websocket.launch http://localhost:11311 89x11'. It shows the user running 'roslaunch rosbridge\_server rosbridge\_websocket.launch'. The output is similar to the first window, indicating logging, disk usage check, and starting the 'roslaunch server' on port 58801. Both windows end with a 'SUMMARY' line.

```
roscore http://daneel:11311/ 89x11
fzhan@daneel:~/Documents/website/rosdash-devel/file$ cd ~
fzhan@daneel:~$ roscore
... logging to /home/fzhan/.ros/log/da8349f6-48d9-11e3-9254-00219b226f0e/roslaunch-daneel-23597.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://daneel:46702/
ros_comm version 1.9.50

SUMMARY

/opt/ros/groovy/share/rosbridge_server/launch/rosbridge_websocket.launch http://localhost:11311 89x11
fzhan@daneel:~$ roslaunch rosbridge_server rosbridge_websocket.launch
... logging to /home/fzhan/.ros/log/da8349f6-48d9-11e3-9254-00219b226f0e/roslaunch-daneel-26195.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://daneel:58801/

SUMMARY
```

# Steps to Create a Dashboard (10)

- 10. Open panel page, and connect with rosbridge;
- 11. Run robots and see

