UNIVERSITY OF NEW SOUTH WALES

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

User manual for UNSW's BellaBot

Warnings

- NEVER connect the bot to an internet-connected network through wifi. Nothing should happen, but there is a risk of the bot bricking itself.

Table of contents

Booting up the BellaBot

To boot up the BellaBot, follow the steps below.
1. Hold down the power button at the top back of the bot for at least 3 seconds, then let go.
2. Wait for the lights to turn on and the operating system to boot up.
3. Once the Android home screen is visible, the bot has been booted up successfully.

Connecting a controller/gamepad

1. To connect and pair a wireless controller or keyboard, make sure that the controlling device is currently disconnected from any other device and that it is on.
2. Swipe down from the top of the interface.
3. Hold down the Bluetooth icon to bring up the Bluetooth settings page.
4. Ensure Bluetooth is on.
5. If your device has already been paired to the BellaBot before, select your device to connect it.
6. If your device has not been paired to the BellaBot before, make sure your device is discoverable, then select "Pair new device", and select your device from the list of available devices.
7. To test if your device is registering, move to Android home then swipe up to access the applications screen. If analog stick or D-pad movement (for controllers) changes the app selected, then the wireless device has been connected successfully.

If the keyboard you are using is wired, then you can also connect the keyboard via one of the USB-A interfaces exposed on the back of the BellaBot face, after you have removed the top back plate.

Using the BellaBotApp

From Android home, swipe up to access the applications screen. Select and open the BellaBotApp.

From the application boot, you can immediately control the BellaBot movement from your connected controller or keyboard, without logging in.

Log in by selecting and typing in the correct log-in details in the username and password fields. This will give you access to the live statistics of the bot, graphs of those statistics over time, a

configurable text page, and an idling page. You can access the idling page by double tapping on any non-input section of the screen (i.e. not in the text input part of the app).

Movement controls using a controller/gamepad

The default maximum speed is set to 0.3. Maximum speeds are increased and decreased at increments of 0.01, and are limited to the range of 0 to 0.5.
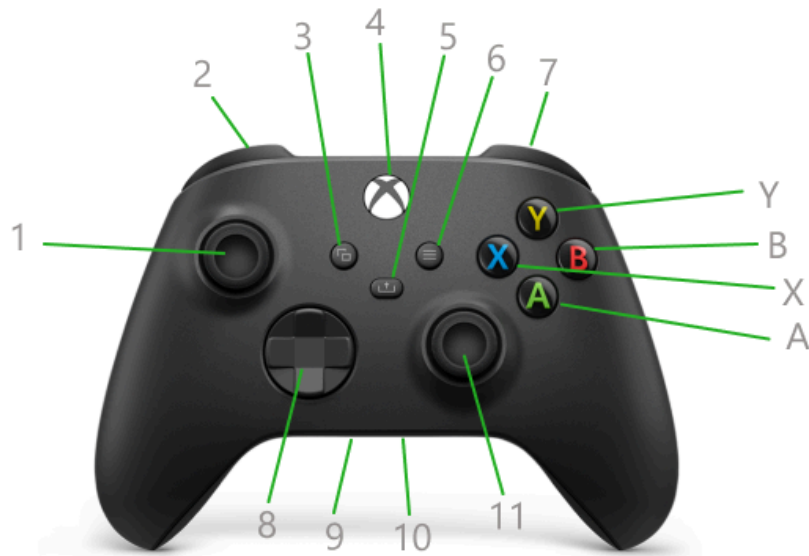
- Left analog stick (1) - forward/backward movement, with the linear speed set based on the position of the stick.
- Right analog stick (11) - left/right rotation, with the rotation speed set based on the position of the stick.
- Left bumper (2) - reduce the max speed. Tap to reduce speed by 0.01. Hold to reduce more at once.
- Right bumper (7) - increase max speed. Tap to reduce speed by 0.01. Hold to increase more at once.
- B button - controller emergency stop enable
- A button - controller emergency stop disable
- X button - clear wheel errors
- Y button - get acceleration data

The rest of the controller buttons and triggers are unassigned.
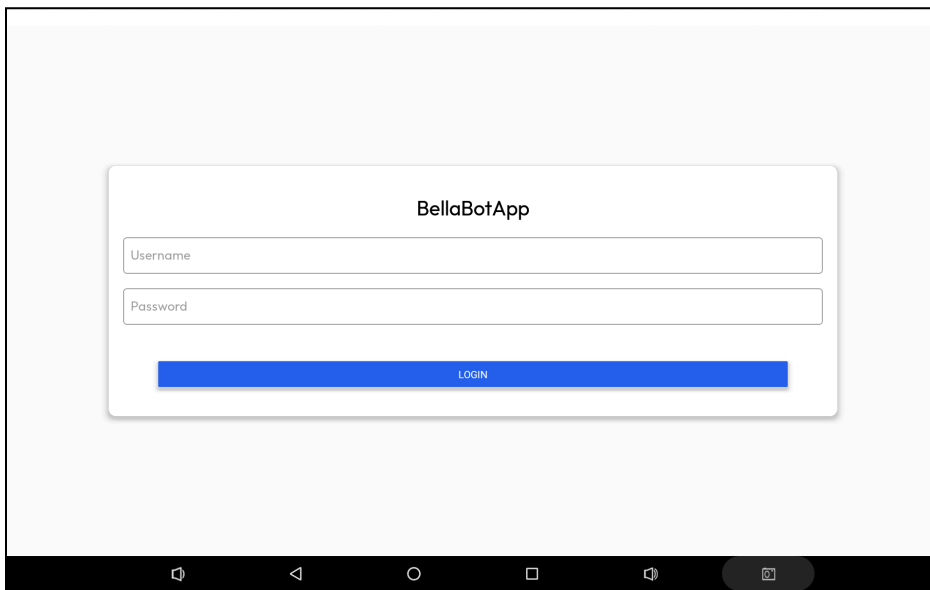
Using the Bellabot Android App

Login

Enter the relevant username and password to gain access to the following pages. Enter the correct username and password, then press the blue LOGIN button to log in.
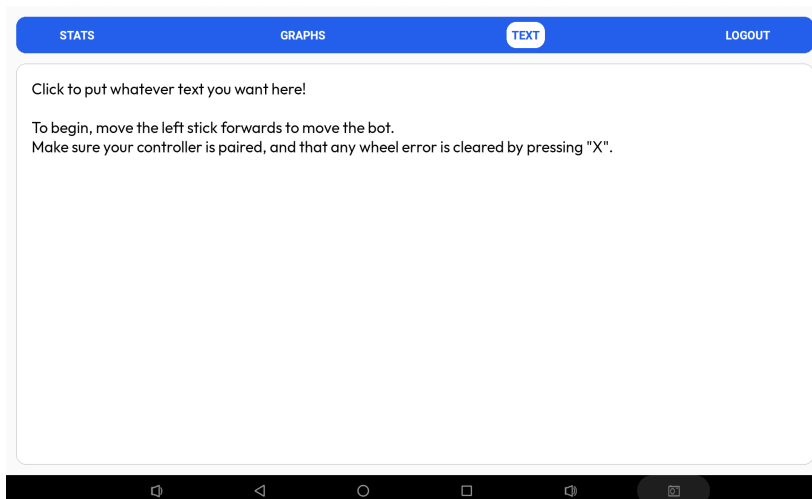The current set username and password is the following.
Username: admin
Password: adminpass

After logging in you will be greeted with the text page of the application with 4 different tabs at the top of the screen to access different pages and functionality.

Stats

Select this tab at the top of the application to access the live statistics page. This shows the current statistics of the BellaBot, which includes the following fields on wheel errors, encoder or wheel speed data, the battery and charge state, and gyroscope (rotational) data.
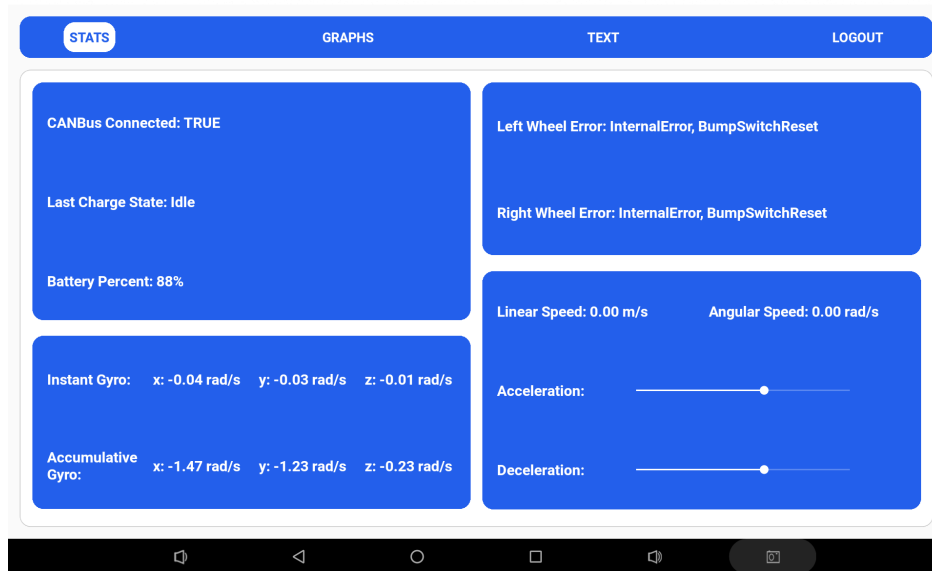

Graphs

Select this tab at the top of the application to access the graphs page. This graphs the statistics of the current active session of BellaBot. The following statistics are graphed:
- Battery percent
- Linear speed
- Angular speed
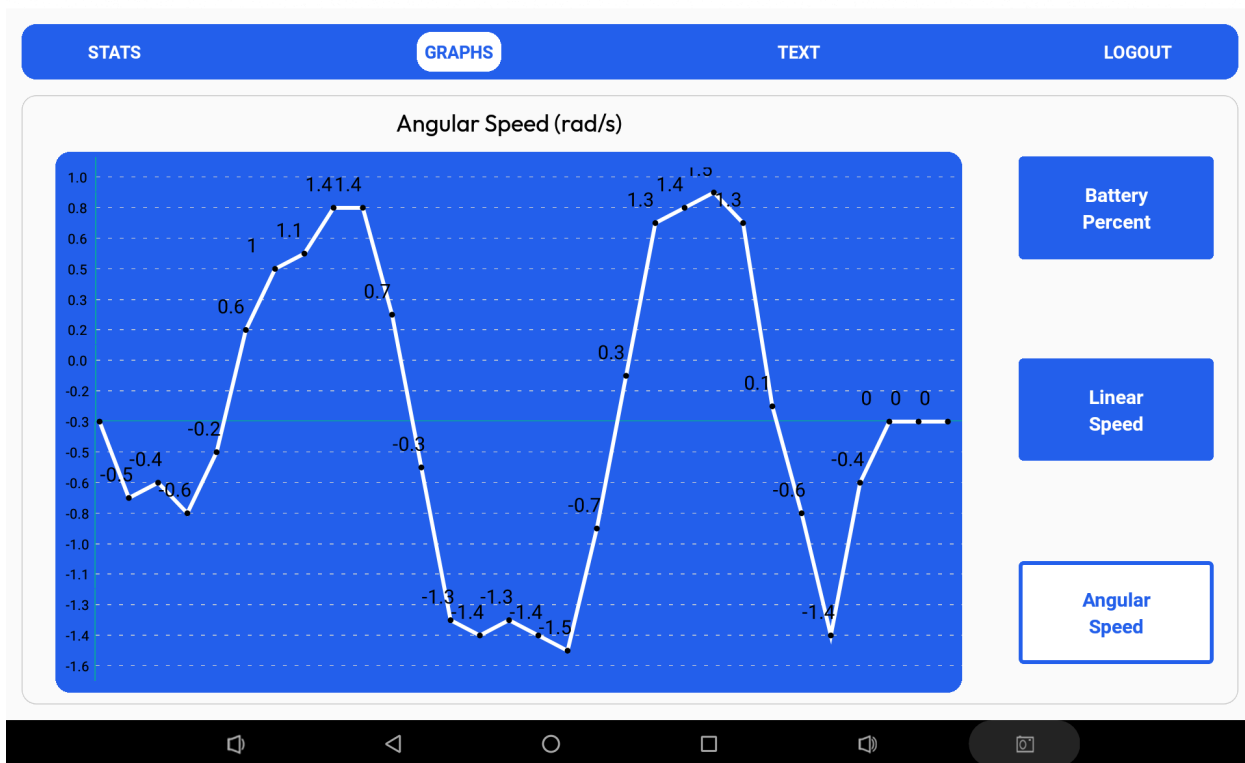
Text

Select this tab at the top of the application to access the text page. This page allows the user to configure the text field to display whatever they wish. To change the text displayed, select the text box in the middle of the screen by tapping it, and using the on-screen or a connected keyboard, type the text to be displayed. Tap out of the text box once complete. This may be used for demonstration or for interaction purposes.
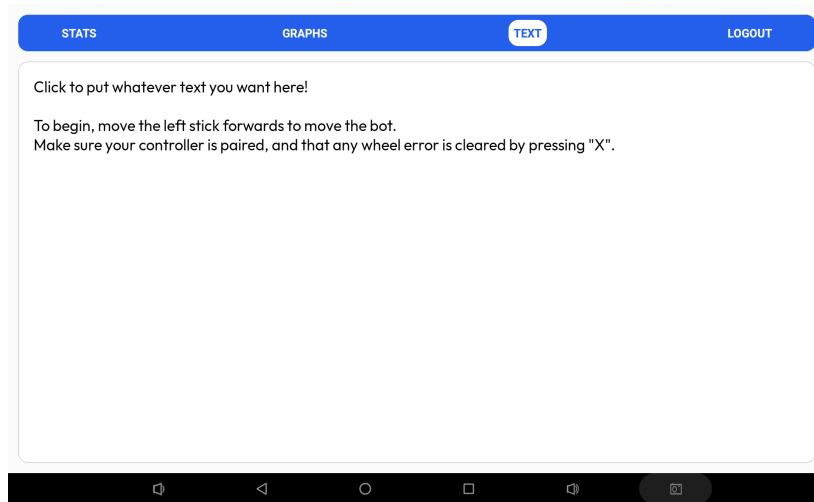
Idling/Faces

To enter the idling state, double tap on any non-text input part of the screen, which will set one of seven random cat faces to fill the entire screen. To exit this state, double tap the screen again.
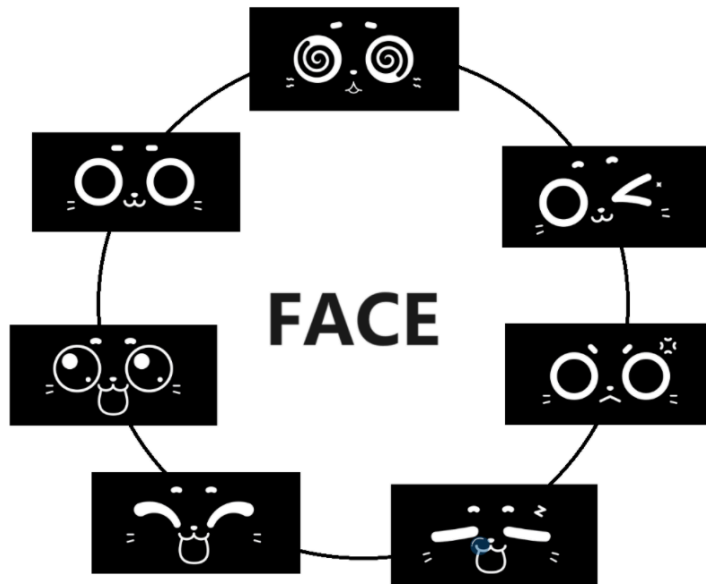
Logout

This button logs you out and removes access to the previous four pages. You must log back in to access live statistics, graphs, the configurable text page, or the idling cat faces.

Maintenance and trouble-shooting

- Make sure the BellaBot has enough charge. Driving it around is particularly hard on the battery.
- If the bot is not moving, it may be caused by one of several issues:
  - The hardware emergency stop (the red button on the top of the bot) is pressed. This will cause a wheel error to be displayed on the Stats screen.
    - Rotate the button clockwise to disengage it.
  - The bottom front of the bot has contacted something hard enough to trigger a wheel error that displays on the Stats screen.
    - Remove the object or move the bot away from the object.
  - There is a miscellaneous wheel error that has not been solved by the above actions.
    - You can see what wheel error is being transmitted on the Stats page.
    - To clear the wheel error and continue movement operation, press A on your controller/gamepad.
  - The software stop is engaged.
    - Press X to disengage the software stop.
  - The controller is not on or is not connected to the BellaBot.
    - See [Connecting a controller/gamepad or keyboard](#) for details on connecting your control device.

Using the ROSBellabot App

Prerequisites

Before running the ROSBellabotApp, ensure the development device you will be using to connect to Bellabot's nodes meets the following prerequisites:

1. **Operating System**: Your development machine should be running Ubuntu 16.04 Xenial Xerus natively or through WSL (Windows Subsystem for Linux).

2. **ROS Kinetic**: ROS Kinetic must be installed on your development machine. This allows interaction with ROS nodes on the Bellabot or other ROS-enabled devices.

Running the App

Step 1: Install ROS on Your Development Machine
- Follow the official ROS installation instructions for Ubuntu 16.04 Xenial Xerus on the device you intend to use for interacting with Bellabot's publishers. This will provide the necessary tools and libraries to interact with ROS nodes.

Step 2: Connect Bellabot and Your Development Machine
- Ensure that the development machine and Bellabot are connected to the same Wi-Fi address. Once connected, note Bellabot's IP address. This will look something like "192.169.1.100".
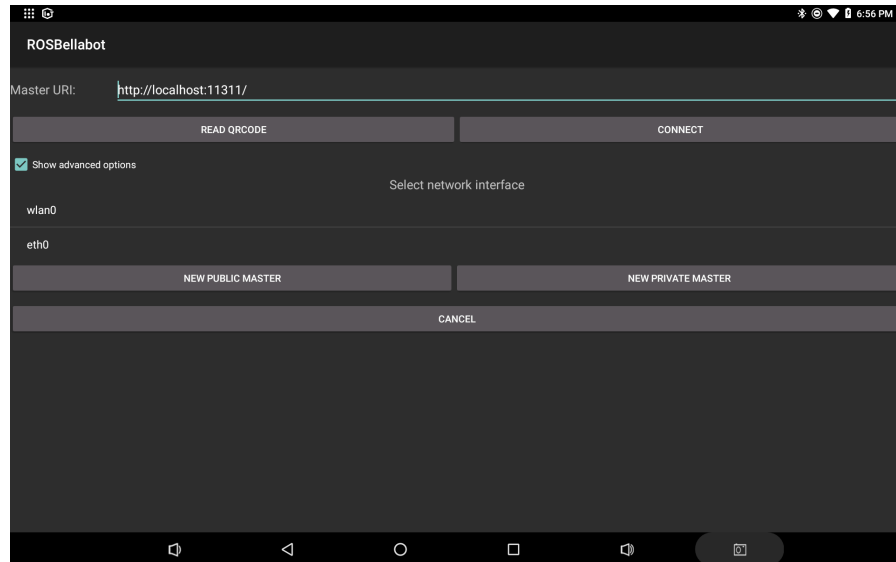
Step 3: Locate the ROSBellabot App on the Android Apps Menu
- Once connected to Wi-Fi, locate and launch the ROSBellabot app from the Apps menu.
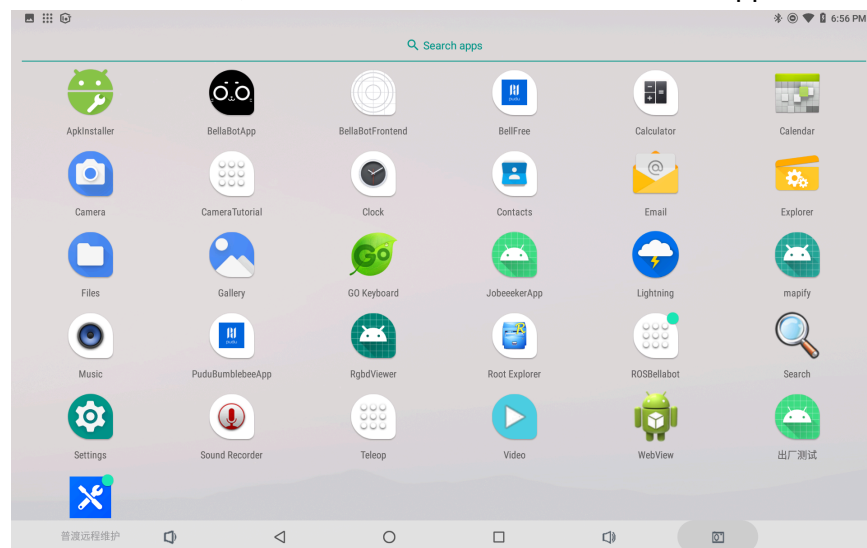
Step 4: Start Bellabot's ROS Master
- Tap '**Show Advanced Options**' to bring up two options that allow running the ROS master on Bellabot (as opposed to connecting to a ROS Master running on an external device).

- Select '**New Public Master**' to begin running the ROS nodes, as well as enable teleoperation. The robot will display a blank white page with the title "ROSBellabot" if all steps have been followed correctly. See Movement controls using a controller/gamepad for movement controls on this app.

Step 5: Connect to Bellabot's ROS Master
- On your development machine, connect to the same WiFi network as Bellabot. Then, open a new Terminal and run `roscore.`

- In a new Terminal window, run `export ROS_MASTER_URI=http://<IP_ADDRESS>:11311`. Replace <IP_ADDRESS> with Bellabot's IP address.
- Finally, run `rostopic list` to see a list of ROS topics published by Bellabot, and `rostopic echo <topic>` to subscribe to a topic and view its contents.

Step 6: Custom ROS Messages
- In a new Terminal window, change to the ROS workspace by running `cd bellabot_ws`.
- Next, run `source devel/setup.bash` to set up the environment variables.
- To begin the custom ROS message publisher, run `rosrun bellabot ros_bellabot.py`.
- Finally, open a new Terminal window, change to the `bellabot_ws` directory and source environment variables. Then, run `rostopic echo /bellabot` to view the custom messages.