Software Setup Manual

1.0 Description

This Software Setup Manual contains detailed information on how to set up the software for a laptop and the Raspberry Pi for our ZelephantBot.

2.0 Setup Guide

2.1 On your remote laptop

1. Installation of Ubuntu 22.04 & ROS2 Humble

   [Things to note before you proceed]
   - Make sure you choose "**Humble**" on the top left corner.

   Follow the instruction shown here:
   https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/#pc-setup

2. Install gazebo (Classic) packages
   1) Install Gazebo 11
   - `sudo apt install gazebo11`
   - `sudo apt install ros-humble-gazebo-ros-pkgs`
   2) Install the Gazebo-dependent Packages according to 3.1.3-2
   - `sudo apt install ros-humble-gazebo-*`
   3) Re-compile the Gazebo packages as per Robotis manual:
   - `cd ~/turtlebot3_ws`
   - `colcon build --symlink-install`
   - `echo 'source ~/turtlebot3_ws/install/setup.bash' >> ~/.bashrc`
   - `source ~/.bashrc`

3. (Optional) Adding aliases and shell variables to the .bashrc to speed up testing
   1) Add the following lines to the .bashrc by running the following commands
   - `export TURTLEBOT3_MODEL=burger`
   - `alias rteleop='ros2 run turtlebot3_teleop teleop_keyboard'`
   - `alias rslam='ros2 launch turtlebot3_cartographer`

`cartographer.launch.py'`

2) Reload the .bashrc by running the following commands

- `source ~/.bashrc`

4. (Optional) Automatic IP Update & SSH Access Setup for RPi

   To streamline the process of accessing your Raspberry Pi (R-Pi) running on the TurtleBot, we automate the IP address retrieval and establish passwordless SSH access to your AWS instance.

   4.1 Generate SSH Key Pair

   1) SSH into the R-Pi from the terminal window:
   - ssh ubuntu@<RPI-IP>
   2) Generate an SSH key pair (press Enter to accept defaults):
   - ssh-keygen
   3) Display the public key on the R-Pi:
   - cat ~/.ssh/id_rsa.pub

   4.2 Add RPi public key to AWS

   1) SSH into your AWS instance:
   - ssh aws
   2) Open autorized_keys:
   - nano ~/.ssh/authorized_keys
   3) Paste the R-Pi's public key and save the file.

   4.3 Configure SSH on the RPi to connect to AWS

   1) On the RPi, create an SSH config file:
   - nano ~/.ssh/config
   2) Add the following (replace IP-address with your AWS instance's public IP):
   - Host aws
   - HostName IP-address
   - User ec2-user

   4.4 Test SSH Access from the RPi to AWS

   - ssh aws

   4.5 Create System Service to Send IP to AWS

   1) Create a new service file:
   - sudo nano /etc/systemd/system/ip2aws.service
   2) Paste the following
   - [Unit]
   - After=network.service

- [Service]
- ExecStart=/home/ubuntu/ip2aws.bash

- [Install]
- WantedBy=default.target

4.6 Create the IP Update Script

1) Create the script file
- nano ~/ip2aws.bash
2) Paste:
- #!/bin/bash
- myIP=`hostname -I | tr -d " "`
- echo $myIP | runuser -l ubuntu -c 'ssh aws "cat - > rpi.txt"'
3) Set permissions:
- chmod 744 ~/ip2aws.bash
- sudo chmod 664 /etc/systemd/system/ip2aws.service
4) Register and enable the service:
- sudo systemctl daemon-reload
- sudo systemctl enable ip2aws.service
5) Reboot
- Sudo reboot

4.7 Verify IP Address is Sent to AWS

On your laptop: ssh aws cat rpi.txt

4.8 Add SSH Aliases on Laptop for Quick Access

1) Edit .bashrc: nano ~/.bashrc
2) Add: alias sshrp='ssh ubuntu@`ssh aws cat rpi.txt`'
3) Reload: source ~/.bashrc

4.9 SetUp Passwordless SSH from Laptop to the RPi

- ssh-copy-id ubuntu@`ssh aws cat rpi.txt`

4.10 Access RPi using shortcut

- sshrp

5. Cloning our laptop programs code

1) Make directory for the laptop_programs
- `mkdir auto_nav_ws`
2) Navigate to the directory where you want the repo
- `cd ~/auto_nav_ws`
3) Clone the Git Repo

- `git clone https://github.com/hyunjinkim1112/r2auto_nav_CDE2310.git`

4) Navigate to r2auto_nav_CDE2310/laptop_programs/frontier_exploration

- `cd ~/auto_nav_ws/r2auto_nav_CDE2310/laptop_programs/frontier_exploration`

5) Build the workspace using colcon build (you may also wish to use symlink install)

- `colcon build`

Now you're ready to run laptop_programs on your RPi.

2.2 On your RPi

1. Installation of Ubuntu 22.04 on your Raspberry PI, configuration of the Raspberry PI, and installation of packages on Raspberry PI

   [Things to note before you proceed]

   - Prepare microSD card and reader before you proceed
   - Follow each and every step of 3.2 SBC Setup (entire) and 3.3 Install Packages on RaspberryPI (except 3.3.2 Rpi Camera)
   - Use your hotspot for WiFi SSID and password when you set up your RPI.

   Follow the instruction here:
   https://emanual.robotis.com/docs/en/platform/turtlebot3/sbc_setup/#sbc-setup

2. (Optional) adding aliases and shell variables to the .bashrc
   1) Add the following lines to the .bashrc
   - `export TURTLEBOT3_MODEL=burger`
   - `alias rosbu='ros2 launch turtlebot3_bringup robot.launch.py'`

   2) Reload the .bashrc:
   - `source ~/.bashrc`

3. Cloning RPi programs from our github
   1) Make directory for the rpi_programs
   - `mkdir auto_nav_ws`

   2) Navigate to the directory where you want the repo
   - `cd ~/auto_nav_ws`

   3) Clone the Git Repo
   - `git clone https://github.com/hyunjinkim1112/r2auto_nav_CDE2310.git`

4) Navigate to r2auto_nav_CDE2310/rpi_programs
- `cd ~/auto_nav_ws/r2auto_nav_CDE2310/rpi_programs`

4. Installing libraries for I2C and AMG8833 sensor
   1) Update and install pip
   - `sudo apt update`
   - `sudo apt install python3-pip -y`

   2) Install Blinka (this brings in busio, board, digitalio, etc.)
   - `pip3 install adafruit-blinka`

   3) Install the AMG88xx library
   - `pip3 install adafruit-circuitpython-amg88xx`

   4) Enable I2C on Raspberry Pi
   - Run: `sudo raspi-config`
   - Go: `Interface Options -> I2C -> Enable`
   - Reboot: `sudo reboot`

   5) (Optional) Test I2C Connection
   - `sudo apt install -y i2c-tools`
   - `i2cdetect -y 1`
   - You should see an address like 69, which is the AMG8833 default I2C address

   Now you're ready to run rpi_programs on your RPi.

3.0 Operating Instructions

1. Turn on ZelephantBot.
2. SSH into the RPi.
   - `sshrp`

   Alternatively, If you haven't done '2.1.4 Automatic IP Update & SSH Access Setup for RPi' of this software setup manual, you can manually ssh into RPi using:
   - `ssh ubuntu@<rpi ip address>`
3. Run the bring up on the **RPi.**
   - `rosbu`
4. Run motor_driver.py and heat_detection_node.py on the **RPi.**

1) Run motor_driver.py
- `python3 motor_driver.py`
2) Run heat_detection_node.py
- `python3 heat_detection_node.py`

5. On your **laptop,** go to the laptop_programs directory and run auto_mapper, coordinator_node.py, r2auto_nav_modified.py.
    1) Run auto_mapper on 1 terminal
    - `ros2 launch auto_mapper auto_mapper.launch.py map_path:=~/map is_sim:=false`
    2) Run coordinator_node.py on the 2nd terminal tab
    - `python3 coordinator_node.py`
    3) Run r2auto_nav_modified.py on the 3rd terminal tab
    - `python3 r2auto_nav_modified.py`
6. To start the autonomous navigation, press ‘r’ on the terminal that is running coordinator_node.py.
7. To terminate the program, press ‘ctrl+C’.
8. If you make any changes to the auto_mapper.cpp program, do remember to colcon build the frontier_exploration directory, or else the changes will not be reflected.