10. AI Large Model Offline Voice Assistant

1. Offline Voice Configuration

Before setting up auto-start, we must ensure the program can work independently offline. This requires modifying the configuration file.

1. **Locate the Configuration File**:

In your project code, find and open the configuration file:

`config/yahboom.yaml`

2. **Modify Configuration Parameters**:

Please check the following parameters in the file and ensure their values match those shown below. If the parameters are missing, please add them.

```
asr:                                    #Voice Node Parameters
  ros__parameters:
    VAD_MODE: 2                         #vad sensitivity
    sample_rate: 16000                  #asr audio sampling rate
    frame_duration_ms:  30              #VAD frame size unit: ms
    use_oline_asr: False                #Whether to use online ASR recognition
(True uses online, False uses offline)
    mic_serial_port: "/dev/ttyUSB0"     #Microphone serial port alias
    mic_index: 0                        #Microphone Index
    language: 'en'                      #asr language

model_service:                          #Model server node parameters
  ros__parameters:
    language: 'en'                      #Large Model Interface Language
    useolinetts: False                  #Whether to use online speech synthesis
(True for online, False for offline)

     # Large Model Configuration
    # llm_platform: 'ollama'              # Optional platforms: 'ollama',
'tongyi', 'spark''qianfan', 'openrouter'
    llm_platform: 'ollama'              # Current large model platform
```

After completing this step, the program is now a purely offline voice service.

2. Create a startup service (Systemd)

Now, we will create a `systemd` service so that `largemodel_control.launch.py` runs automatically at system startup.

2.1 Creating a Startup Script

To ensure `systemd` correctly loads the ROS2 environment, the best practice is to create a simple `bash` script to encapsulate our startup command.

1. **Create the script file**:

In the directory (`~/yahboom_ws/src/largemodel/`), create a file named `start_largemodel.sh`.

```
vim ~/yahboom_ws/src/largemodel/start_largemodel.sh
```

2. **Write the script content**:

Copy and paste the following content into the script file.

```
#!/bin/bash

# Source ROS2 Humble environment
source /opt/ros/humble/setup.bash

# Source Yahboom workspace environment
source /home/sunrise/yahboom_ws/install/setup.bash

# Start the largemodel control script
ros2 launch largemodel largemodel_control.launch.py
```

Important Note: Please ensue that `/home/sunrise/` in the script is replaced with your own user's home directory path.

3. **Save and Exit**
4. **Grant Script Execution Permissions**:

```
chmod +x ~/yahboom_ws/src/largemodel/start_largemodel.sh
```

2.2 Creating a Systemd Service File

This is the most crucial step. We will tell the system that we have a new service that needs to be managed.

1. **Create the service file:**

You need `sudo` privileges to create this file.

```
sudo vim /etc/systemd/system/largemodel.service
```

2. **Write the service configuration:**

Copy and paste the following content into the service file.

```
[Unit]
Description=Robot Service
After=network.target sound.target graphical.target multi-user.target
Wants=network.target sound.target graphical.target multi-user.target
```

```
[Service]
Type=simple
User=sunrise
Group=sunrise
Environment=DISPLAY=:0
Environment=XDG_RUNTIME_DIR=/run/user/1000
Environment=PULSE_SERVER=unix:/run/user/1000/pulse/native
SupplementaryGroups=audio video
ExecStartPre=/bin/sleep 10
ExecStart=/home/sunrise/yahboom_ws/src/largemodel/start_largemodel.sh
Restart=on-failure
StandardOutput=journal
StandardError=journal

[Install]
WantedBy=multi-user.target
```

- `ExecStart` The path in the command must be exactly the same as the path to the startup script to be executed.

3. **Save and exit**.

2.3 Managing and Debugging the Service

Now that your service has been created, we need to get `systemd` to load it and set it to start automatically on boot.

1. **Reload the `systemd` daemon** so that it reads the service file we just created:

```
sudo systemctl daemon-reload
```

2. **Set the service to start automatically on boot**:

```
sudo systemctl enable largemodel.service
```

3. **Start the service immediately**:

```
sudo systemctl start largemodel.service
```

4. **Check the service status**:

This is the most important command to verify that the service is running successfully.

```
sudo systemctl status largemodel.service
```

- If you see `Active: active (running)`, congratulations, the service has started successfully!
- If the status is `failed` or something else, please continue to the next step for debugging.

5. **View service logs (essential for debugging)**:

If the service fails to start, you can view all real-time logs generated by the `ros2 launch` command using the following command. This is crucial for locating the problem.

```
journalctl -u largemodel.service -f
```

After completing all the above steps, the purely offline `largemodel` voice service will now start automatically every time you boot up.