8. AI Large Model Online Voice Assistant

8. AI Large Model Online Voice Assistant
    1. Online Voice Configuration
    2. Create Boot Auto-start Service (Systemd)
        2.1 Create a Startup Script
        2.2 Create Systemd Service File
        2.3 Manage and Debug Service

1. Online Voice Configuration

Before setting up auto-start, we must ensure that the program itself can work independently in online mode. This needs to be done by modifying configuration files.

1. **Locate 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 parameters do not exist, please add them.

```
asr:                                    # Voice node parameters
  ros__parameters:
    VAD_MODE: 2                         # VAD sensitivity
    sample_rate: 16000                  # ASR recording audio sample rate
    frame_duration_ms:  30              # VAD frame size in ms
    use_oline_asr: True                # Whether to use online ASR
recognition (True for online, False for offline)
    mic_serial_port: "/dev/ttyUSB0"     # Microphone serial port alias
    mic_index: 0                        # Microphone index
    language: 'zh'                      # ASR language

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

     # Large model configuration
    # llm_platform: 'openrouter'        # Available platforms: 'ollama',
'openrouter'
    llm_platform: 'openrouter'              # Taking openrouter as example
here
```

   - `use_oline_asr` and `useolinetts` must be set to True.
   - `llm_platform` must be set to `openrouter`.

     With these settings, everything will be online.

3. **Save the file** and **recompile** the project to apply changes:

```
cd ~/yahboom_ws
colcon build
source install/setup.bash
```

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

2. Create Boot Auto-start Service (Systemd)

Now, we will create a `systemd` service to make `largemodel_control.launch.py` run automatically when the system starts.

2.1 Create a Startup Script

To enable `systemd` to properly load the ROS2 environment, the best practice is to create a simple `bash` script to wrap our startup command.

1. **Create 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 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 largemodel control script
   ros2 launch largemodel largemodel_control.launch.py
   ```

   Important Note: Please ensure you replace `/home/sunrise/` in the script with your own user home directory path.

3. **Save and Exit**

4. **Grant Script Execution Permission**:

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

2.2 Create 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 Service File**:
   You need to use `sudo` privileges to create this file.

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

2. **Write Service Configuration**:
   Copy and paste the following content completely 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
```

   !!! Extremely Important !!!

   - Please ensure that the paths in `WorkingDirectory` and `ExecStart` exactly match your actual paths.

3. **Save and Exit**.

2.3 Manage and Debug Service

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

1. **Reload `systemd` daemon** to let it read our newly created service file:

```
sudo systemctl daemon-reload
```

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

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

3. **Start service immediately**:

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

4. **Check service status**:
   This is the most important command to verify whether 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 other, 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 through the following command, which is crucial for locating problems.

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

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