

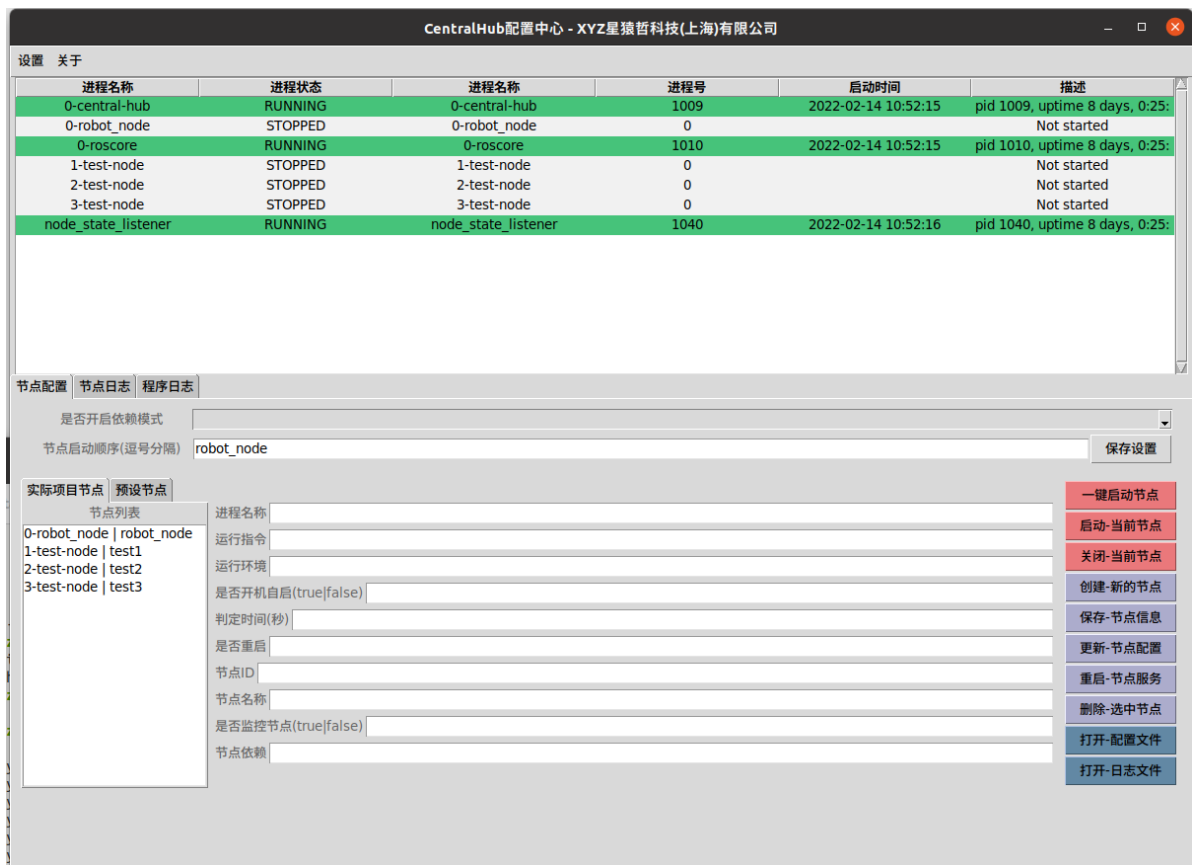
xyz-supervisor用户手册

文档更新日期：2022-07-20

- 一、基本介绍
 - 1. 关于初始节点的构成
- 二、如何安装
 - 1. xyz-studio依赖安装
 - 2. 使用xyz-csc安装软件
 - 3. 使用xyz-apt安装软件
- 三、如何配置
 - 1. 配置节点
 - 2. 配置central hub参数
 - 3. 添加新的节点
 - 4. 删除无用的节点
 - 5. 修改节点配置
 - 6. 配置节点启动顺序
 - 7. 节点启动失败的问题排查
- 四、如何启用
 - 1. 重新加载节点配置
 - 2. 重启supervisor服务
- 五、节点配置步骤总结
- 六、节点参数说明
- 七、注意事项
 - 1. 更新版本后配置文件未同步更新
 - 2. 配置带界面的节点程序
- 八、常见问题
 - 1. 什么是节点监听？
 - 2. 为什么有的节点配置参数前面有下划线，比如 `_nodeid` ？
 - 3. 为什么新建一个节点并且到central hub添加到启动顺序中后，HMI中依然没有显示？
 - 4. 【重启节点服务】后central_hub_gui界面里节点列表空了，9001端口的网址也打不开了？
- 九、其他操作
 - 1. 查看supervisord.conf配置
 - 2. 在central_hub_gui中启动、停止节点
 - 3. 在central_hub_gui中查看节点日志
 - 4. 在软件界面中打开本文档
 - 5. 查看当前软件版本
 - 6. 调整节点日志的字体大小
- 十、问题反馈

一、基本介绍

Supervisor是一款采用Python编写、运行在Linux/Unix系统上的进程管理工具。该工具提供了一系列管理进程的功能。例如，启动进程、停止进程、查看进程日志、重启进程、分组管理进程、进程开机自启动、RPC接口等。但是由于Supervisor不能控制进程的启动顺序，也不能知道进程初始化完成的时间，因此需要Central Hub加以辅佐。并且，**central hub作为一个特殊的节点运行在supervisor中**。使用central hub需要下载xyz-supervisor的deb包进行安装。为了方便配置启动节点，xyz-supervisor安装包中还提供了GUI界面，采用python + tkinter编写。启动方式为：在终端输入 `central_hub_gui` 回车即可打开，启动后你将看到如下界面：



1. 关于初始节点的构成

打开【节点配置中心】界面，有三个初始节点在保持RUNNING的运行状态，其他为用户自定义节点。初始节点分别是

- ① 0-central-hub
- ② 1-roscore
- ③ node_state_listener

下面解释这三个节点的作用

0-cental-hub

该节点就是运行在supervisor中的一个特殊节点，开机自启。提供了节点管理功能的一些扩展，如控制节点的启动顺序，监听节点等功能。其他程序控制节点启动一般也是通过central hub进行控制。调用代码示例如下：

```
"""
HubClient提供了获取节点信息、启动节点、停止节点、重启节点等功能

需要注意：
roscore、node_state_listener这些节点不由central_hub控制启动
central_hub一般只控制【节点启动顺序】中注册的节点
"""

from xyz_central_hub.client import HubClient
cli = HubClient()
cli.start_node('robot_node')
```

1-roscore

roscore开机自启，如果在其他地方已经启动了roscore节点，那么该节点会因为端口占用问题启动失败。

node_state_listener

节点状态的事件监听程序，可以监听节点的运行状态的变更，如当节点突然异常退出时，会推送停止信号到central hub程序。配置参数如下：

```
[eventlistener:node_state_listener]
command=/usr/bin/python -u
/home/xyz/xyz_app/central_hub/script/node_state_listener.py
events=PROCESS_STATE
process_name=%(program_name)s
priority=1
autostart=true
autorestart=unexpected
startsecs=1
;startretries=1
user=xyz
stopasgroup=true
;
stdout_logfile=/home/xyz/xyz_log/central_hub/nodes/%(program_name)s.log
stdout_logfile_maxbytes=10MB
stdout_logfile_backups=2
stdout_capture_maxbytes=10MB
stdout_events_enabled=false
;
stderr_logfile=/home/xyz/xyz_log/central_hub/nodes/%(program_name)s.log
stderr_logfile_maxbytes=10MB
stderr_logfile_backups=2
stderr_capture_maxbytes=10MB
stderr_events_enabled=false
```

二、如何安装

公司的系统环境中默认已经安装了supervisor，因此实际安装时只需要再下载一个xyz-supervisor安装包。另外，如果工控机中下载xyz-studio，那么会将xyz-supervisor一同下载。

1. xyz-studio依赖安装

如果工控机中已经下载xyz-studio，那么就会将**xyz-supervisor**作为**依赖软件一同下载**，之后便不再需要额外下载一次xyz-supervisor了。xyz-studio请参考该软件的相关手册，此处不作额外说明。

2. 使用xyz-csc安装软件

1. 打开xyz-csc
2. 进入【软件安装/卸载】模块
3. 搜索xyz-supervisor，在点击【安装】xyz-supervisor软件



3. 使用xyz-apt安装软件

1. 更新索引源：`sudo xyz_apt update`
2. 安装软件包：`sudo xyz_apt install xyz-supervisor`
3. 登录 [<http://127.0.0.1:9001>] 或者在终端输入 `central_hub_gui`，如页面正常打开，则说明安装成功。

三、如何配置

1. 配置节点

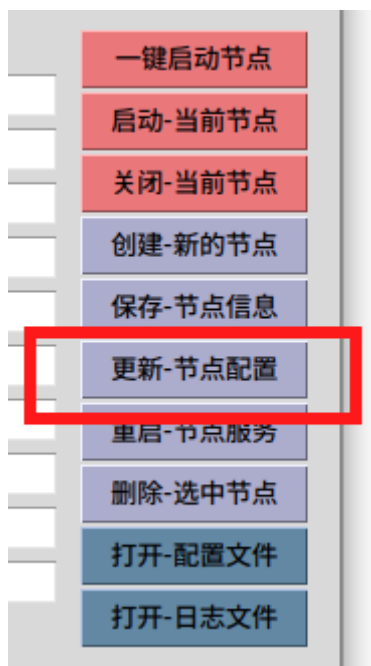
项目类型用简写的字母表示。如 `rebin`指播种站项目，`pp`指拣选站项目，`dpt`指拆码垛项目，`ind`指工业项目。

`central_hub_gui` 界面操作方式

打开`central_hub_gui`，在【节点配置】中点击预设节点，选择一个项目类型，例如【拆码垛（预设）- `dpt`】，点击【拷贝预设节点至实际项目中】，确认后即可在实际项目节点中查看。



需要注意的是，拷贝预设节点后需要再点击【更新-节点配置】后，监控中的节点配置才会更新生效。之后修改了节点后需要更新节点配置也是同理。



文件操作方式

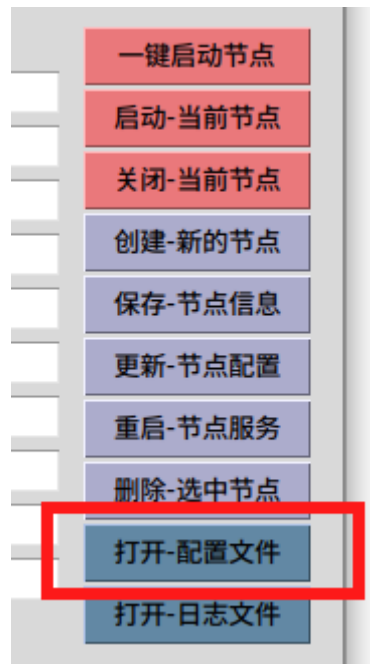
1. 打开目录：`/home/xyz/xyz_app/central_hub`。所有和central hub相关的配置文件都在该目录下。
2. 选择相应项目类型的节点配置文件。如使用拆码垛的预设节点配置，其文件路径为 `/home/xyz/xyz_app/central_hub/node/dpt/nodes.ini`。将该配置文件拷贝至 `/home/xyz/xyz_app/central_hub/node/` 目录，覆盖原来的node.ini文件
3. 打开 `/home/xyz/xyz_app/central_hub/node/nodes.ini` 文件，在文件中进行节点配置。一个 `[program:xxx]` 块就是一个节点。

2. 配置central hub参数

central hub的相关参数基本不需要修改，唯一需要修改的就是节点启动顺序 `boot_sequence`。默认的central hub节点配置参数如下：

```
[central_hub_app]
supervisor_username=xyzrobot
supervisor_password=robot2022
supervisor_address=127.0.0.1:9001
hub_server_address=127.0.0.1:9004
node_server_address=127.0.0.1:10001
node_error_url=http://127.0.0.1:7002/api/notify/node_error
dependency_mode=false
; 节点启动顺序，节点之间使用逗号分隔
boot_sequence=robot_node
```

可在central_hub_gui中点击【打开-节点配置】查看，或者也可以直接打开文件路径 `/home/xyz/xyz_app/central_hub/node.ini` 进行查看修改。



```

1 [central_hub_app]
2 supervisor_username=xyzrobot
3 supervisor_password=robot2021
4 supervisor_address=127.0.0.1:9001
5 hub_server_address=127.0.0.1:9004
6 node_server_address=127.0.0.1:10001
7 node_error_url=http://127.0.0.1:7002/api/notify/node_error
8 dependency_mode=false
9 boot_sequence=robot node
10
11
12 [program:0-robot_node]
13 command=/opt/ros/noetic/bin/rosrun xyz_robot xyz_robot_node
14 process_name=%(program_name)s
15 priority=1
16 autostart=false
17 autorestart=false
18 startsecs=1
19 startretries=1
20 user=xyz
21 stopasgroup=true
22 stdout_logfile=/home/xyz/xyz_log/central_hub/nodes/%(program_name)s.log
23 stdout_logfile_maxbytes=10MB
24 stdout_logfile_backups=2
25 stdout_capture_maxbytes=10MB
26 stdout_events_enabled=false
27 stderr_logfile=/home/xyz/xyz_log/central_hub/nodes/%(program_name)s.log
28 stderr_logfile_maxbytes=10MB
29 stderr_logfile_backups=2
30 stderr_capture_maxbytes=10MB
31 stderr_events_enabled=false
32 ; central hub config
33 _nodeid=robot_node
34 _nodename=robot driver 2.0
35 _monitor=true
36 _popup=false
37 _dependency=

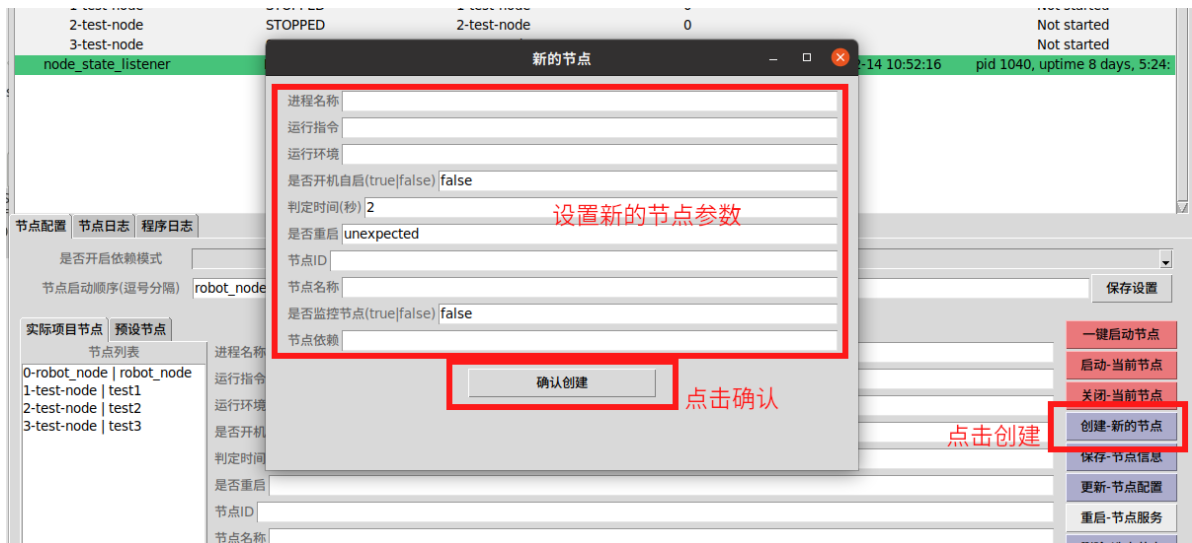
```

3. 添加新的节点

如果不需要添加新的节点，此步骤可跳过

如果项目预设中缺少需要节点，这时就需要手动创建一个新的节点。操作方式如下：

(1) 点击【创建-新的节点】，你会看到下图所示界面：



(2) 设置新节点的参数。参数设置方式如下：

- **进程名称** - 如 `10-new_node`，不可与supervisor节点列表中的其他进程名称重复，否则会导致配置失败
- **运行指令** - 运行脚本的指令，可参考节点在命令行运行时的命令，但是最好把相对路径改为绝对指令，如在终端输入的 `roslaunch` 改为 `/opt/ros/noetic/bin/roslaunch`；如果命令中使用了环境变量，如 `$CODE_BASE`，则需要修改为其实际值 `/home/xyz/xyz_app`，切记
- **运行环境** - 部分节点可能需要在程序运行时读取环境变量，但是supervisor的环境变量不会读取 `/home/xyz/.bashrc` 文件中的环境变量。因此需要将节点需要用的环境变量再配置到这个节点中
- **是否开机自启** - true或者false
- **判定时间(秒)** - 如果这个节点可以启动达到判定的时间长度且不停止，那么认为这个节点启动成功。
- **是否重启** - true或者false，

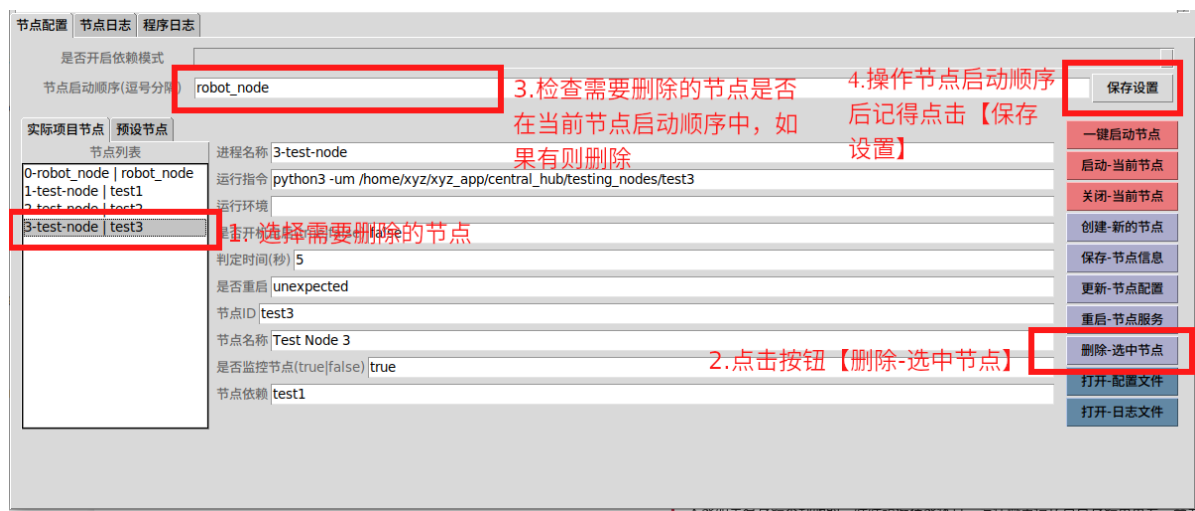
- **节点ID** - 节点的唯一编号，不可与其他节点ID重复
- **节点名称** - 方便在HMI上显示名称，支持中文显示
- **是否监控节点(true|false)** - 即监控节点是否成功启动。节点启动达到判定时间依然没有退出，也不认为启动成功，只有在节点主动推送了一个启动成功信号给central_hub后，才认为该节点启动成功，
- **节点依赖** - 目前项目中暂时不需要了，可不填

(3) 点击【确认创建】，之后会在实际项目节点中新增该节点。如果节点参数中有参数设置错误，更新节点配置的时候，可能会导致supervisor服务异常停止。这时需要

在 `/home/xyz/xyz_app/central_hub/node.ini` 节点配置文件中手动检查参数的有效性。

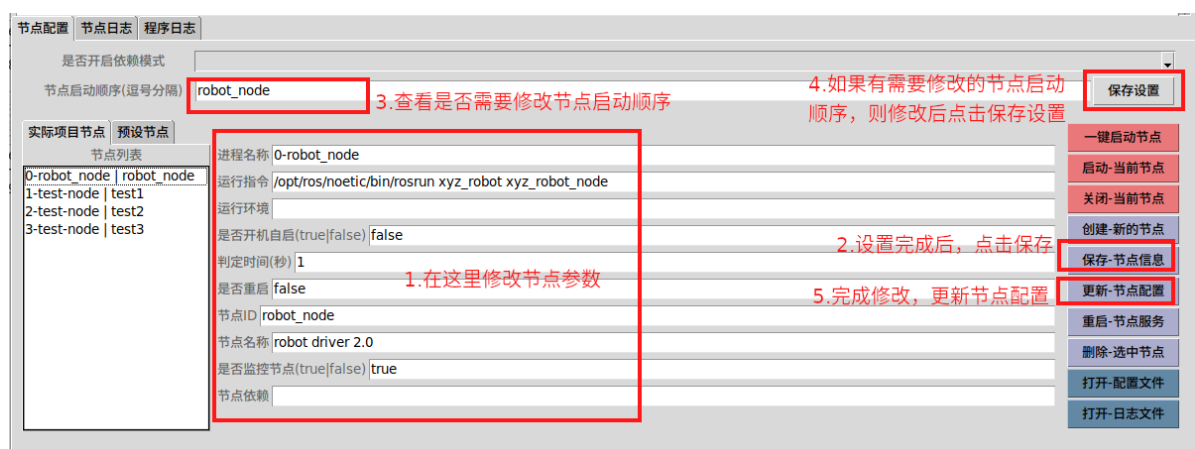
4. 删除无用的节点

如果一个节点在项目中不需要用到，可以将这个节点删除。操作方法如下：



5. 修改节点配置

修改节点配置一般只需要在界面中设置后点击【保存-节点信息】，再点击【更新-节点配置】即可。如果需要修改节点启动顺序，那么中间再修改节点启动顺序。



6. 配置节点启动顺序

文件操作方式

打开 `/home/xyz/xyz_app/central_hub/nodes.ini` 文件，找到 `boot_sequence` 字段。添加节点顺序。（每次修改节点后要记得确认节点顺序是否发生变动）

界面操作方式

终端输入 `central_hub_gui` 打开图形界面，在【节点配置】标签页中修改找到【节点启动顺序（逗号分隔）】进行修改，修改完成后**记得保存并重启central hub**



7. 节点启动失败的问题排查

当启动一个节点时可能会出现节点启动失败的情况，这时候就需要查看节点配置和程序启动的日志信息。

查看节点配置

可以在 `central_hub_gui` 界面中的【节点配置】标签页中查看，也可以点击【打开-节点配置】查看详细的节点配置信息。

查看节点日志

可以在 `central_hub_gui` 界面中点击【节点日志】，再在【节点状态列表】里点击想查看的节点，这时候就会有日志信息显示在【节点日志】中

四、如何启用

1. 重新加载节点配置

当我们更新并保存了一些节点配置后，节点管理服务中的节点配置不会主动更新，这时就需要我们重新加载一次节点配置。**这种加载方式并不会关闭supervisor服务**

```
supervisorctl update
```

2. 重启supervsior服务

（注意：该操作会重启supervisor服务，之前已打开的节点将全部被关闭后再重新打开。）如果需要修改 `supervisord.conf` 配置，则必须重启supervisor服务才能生效，执行指令如下：

```
sudo systemctl restart xyz_autostart.service
```

或者

```
supervisorctl reload
```

五、节点配置步骤总结

配置整站需要的软件节点（以拆码垛项目为例）

1. 在终端输入 `central_hub_gui` 打开节点配置界面。
2. 导入项目预设到实际项目中
3. 增加、删除、修改需要变动的节点配置，并保存节点设置
4. 设置节点启动顺序
5. 更新节点配置。

六、节点参数说明

节点配置过程中涉及很多参数设置。我们以一个节点示例来举例：

```
[program:4-robot_node]
environment =
command = /usr/bin/python -u
/home/xyz/xyz_app/central_hub/script/start_robot_node.py
process_name = %(program_name)s
priority = 40
autostart = false
autorestart = false
startsecs = 2
startretries = 1
user = xyz
stdout_logfile = /home/xyz/xyz_log/central_hub/nodes/%(program_name)s.log
stdout_logfile_maxbytes = 10MB
stdout_logfile_backups = 2
stdout_capture_maxbytes = 10MB
stdout_events_enabled = false
stderr_logfile = /home/xyz/xyz_log/central_hub/nodes/%(program_name)s.log
stderr_logfile_maxbytes = 10MB
stderr_logfile_backups = 2
stderr_capture_maxbytes = 10MB
stderr_events_enabled = false
_nodeid = robot_node
_nodename = Robot Driver 2.0
_monitor = true
_popup = false
_dependency =
_description =
```

先介绍几个比较重要的节点参数：

- **environment**：设置节点的单独环境变量
- **command**：启动这个节点的指令。该指令最好使用绝对路径，因为supervisor在运行时其环境变量不是xyz用户下的环境变量。而是单独在supervisord.conf配置文件中设置了环境变量

- **autostart**: 是否开机启动（随supervisor服务启动后启动）
- **autorestart**: 是否自动重启
- **startsecs**: 该节点启动成功的判定时间，比如设置2，那么就是指该节点需要启动运行2秒不退出后，会认为这个节点启动成功
- **_nodeid**: 节点编号，供central hub使用。节点编号作为唯一标示在central hub中用于识别节点。目前robot_node节点的节点ID固定为 robot_node，其他节点的节点ID非固定。
- **_nodename**: 节点名称，供central hub使用。该名称主要是为了方便用户看，也可设置为中文
- **_monitor**: 节点启动成功的信号监听。数值可设置true或false，如果为true，就开启对该节点的监听。即当启动该节点后，需要该节点向10001端口的socket服务推送一个启动成功的信号。**需要注意推送的启动成功信息中node_id这个字段的字段名是 node_name**，推送信息如下：

```
{
  "code": 0,
  "msg": "",
  "node_name": "node_id",
  "timestamp": 1644569370365
}
```

以下参数一般不需要额外改动，使用默认即可。另外，**stderr_logfile_maxbytes**等参数此处不作额外说明，主要用于设置日志的大小和备份数量等。

- **process_name**：在supervisor中显示的进程名称，%(program_name)s是指使用 [program:4-robot_node] 中的4-robot_node
- **priority**：启动优先级，数值越大优先级越低。即使设置了也没有太大作用
- **user**：进程用户名，默认使用xyz
- **_dependency**：节点依赖关系，经过各软件迭代，节点依赖的功能已经很少会被使用，目前没有使用需求。
- **_description**：节点作用描述。

七、注意事项

1. 更新版本后配置文件未同步更新

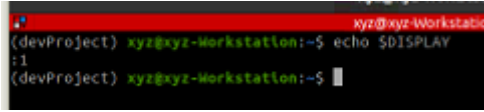
由于xyz-supervisor下载安装后会在 /home/xyz/xyz_app 目录下生成central_hub文件，当更新xyz-supervisor软件的时候，其配置文件不会随着软件卸载而被卸载。因此，当需要使用更新版本的配置文件时，需要首先删除或移除原来的文件夹，路径为 /home/xyz/xyz_app/central_hub，再更新软件。

补充：如果软件已经更新到了最新版本，但是忘记删除原来的配置文件，这时需要使用最新版本的配置文件该怎么办？方法二选一。① 卸载xyz-supervisor后删除配置文件，再重新安装最新版本。②将xyz-supervisor降低一个版本，再删除配置文件后更新到最新版本。

2. 配置带界面的节点程序

首先在终端使用指令`echo $DISPLAY`查看当前终端中`DISPLAY`的值，之后再在带界面的节点配置中，设置其环境变量。

比如，一般`echo $DISPLAY`返回的值为`:1`



```
xyz@xyz-Workstation:~$ echo $DISPLAY
:1
xyz@xyz-Workstation:~$
```

顺带一提，部分GUI程序中，还有一些内部环境变量的参数配置，可以通过这些参数配置GUI的使用或不使用

八、常见问题

1. 什么是节点监听？

节点启动有两种模式，一种是普通模式，另一种是节点监听模式。它们通过不同的标准来判断节点是否启动成功。普通模式下，根据设置的启动时间来判断节点是否成功启动。即，如果程序成功运行N秒并且中途没有退出，那么认为该节点启动成功。但是这种模式不能很好的运用在socket连接通信中。比如，socket服务启动后，虽然成功启动了，但是这个时候并没有与客户端建立连接，因此需要一个节点中的socket服务在与客户端建立连接后回调信息，告知central_hub程序服务端已经与客户端建立连接。因此，就有了节点监听模式。

比较典型的就是robot_node节点，为了确保robot_node可以真正与机器人建立连接，因此我们需要启用节点的监听功能。robot_node节点启动后，会尝试等待与机器人建立通信，这个过程节点一直处于RUNNING状态。当与机器人通信建立后，便会发送一个请求到central_hub的节点服务中，告知节点服务，robot_node节点启动成功。但是需要注意的是，robot_node启动回告功能也需要设置。配置文件的路径在`/home/xyz/xyz_app/robot_config/robot_config.yml`。

由此延伸导致可能造成的误解，如：

- 为什么节点界面中robot_node的状态是RUNNING，但是在studio、HMI中的机器人节点依然没有启动？因为程序虽然启动了，但是central hub没有监听到robot_node这个节点发送的节点启动成功信息。

2. 为什么有的节点配置参数前面有下划线，比如 `_nodeid` ？

这时提供给central hub程序使用的参数。目前带下划线的配置参数有`_nodeid`，`_nodename`，`_monitor`，`_popup`，`_dependency`，`_description` 6个。带上了下划线是因为这些参数本身并不是supervisor的参数，而是在central_hub程序中需要用到。加上下划线方便区分这些自定义添加的参数字段。supervisor控制节点启停，提供基本的节点管理功能，central_hub程序相当于在这个基础上又进行的一次功能拓展。

3. 为什么新建一个节点并且到central hub添加到启动顺序中后，HMI中依然没有显示？

可能原因如下：

- 没有更新节点配置
- 更新了节点配置，但是并没有重启central hub，导致central hub的节点列表没有更新
- central hub未启动
- HMI因为接口、网络等问题导致数据请求失败

4. 【重启节点服务】后central_hub_gui界面里节点列表空了，9001端口的网址也打不开了？

可能是因为配置的节点参数出错了，导致将supervisor服务关闭后，再次加载节点参数时出错，致使节点不能正常开启。

九、其他操作

1. 查看supervisord.conf配置

点击界面左上角菜单栏按钮【设置】>【Supervisord参数设置】

2. 在central_hub_gui中启动、停止节点

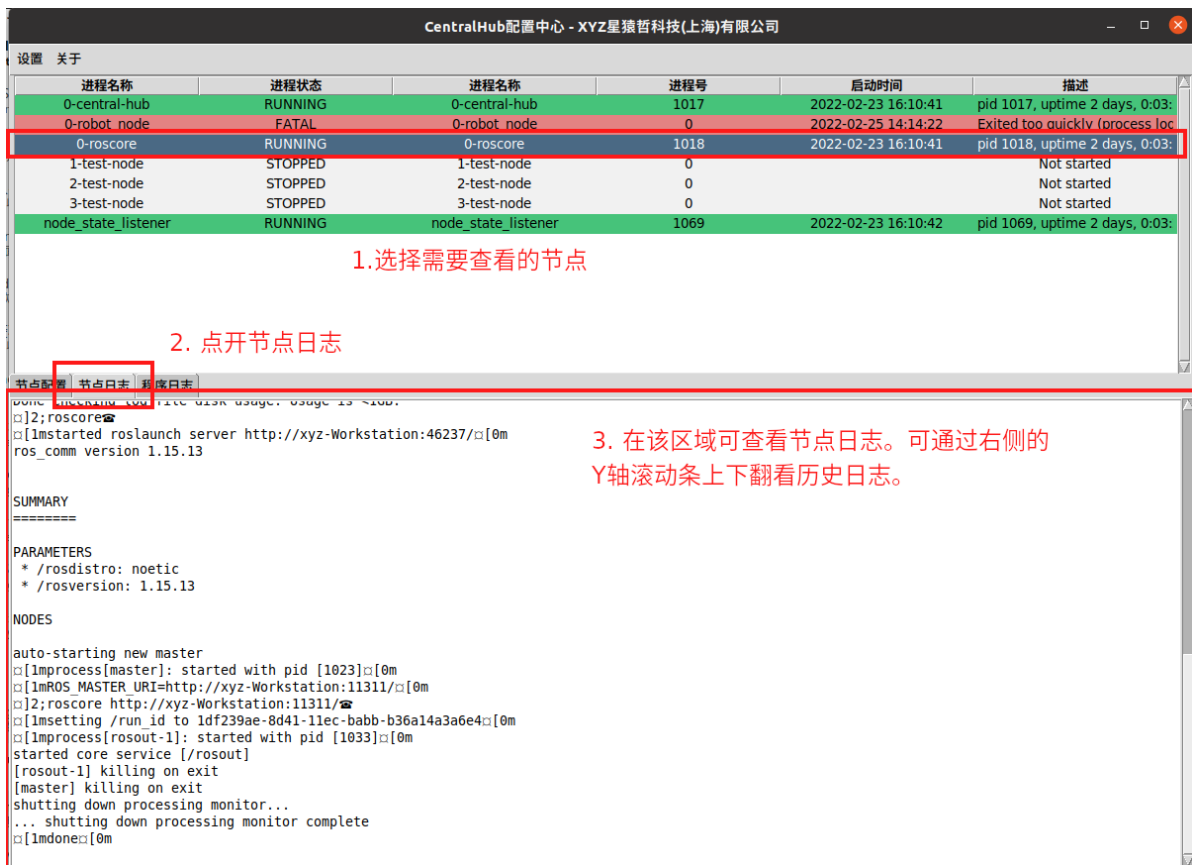
除去上述提供的节点配置操作，该 GUI 还提供了其他的功能，主要是用于节点的查看和调试。如界面上半部分的节点监控中可以控制节点的启动和停止：

- 启动进程：组合键【ctrl+S】
- 停止进程：组合键【ctrl+T】

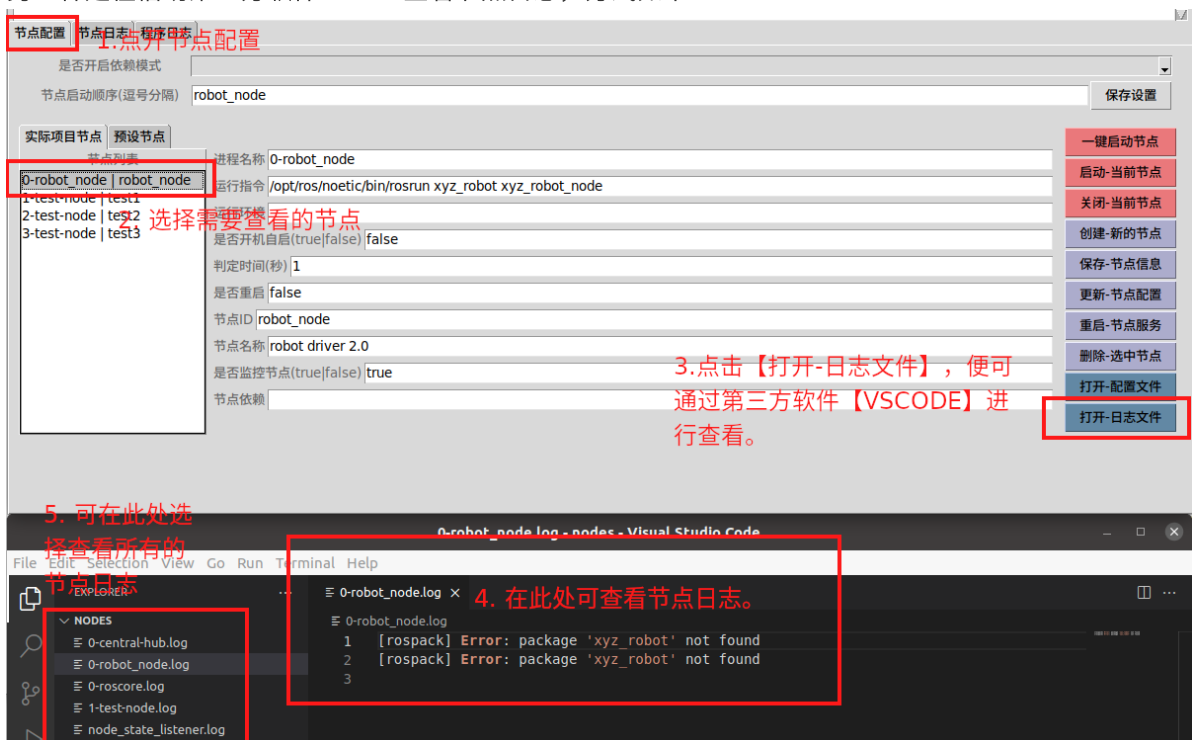
3. 在central_hub_gui中查看节点日志

查看节点日志有两种方式，一种是在【节点状态监控栏】中选择需要查看的节点，然后在界面中的【节点日志】中可查看。

注：有时会出现日志不显示的情况，可能和程序的输出流没有设置及时刷新的原因有关。



另一种是在借助第三方软件vscode查看节点日志，方式如下：



4. 在软件界面中打开本文档

点击界面左上角菜单栏按钮【关于】>【使用说明-点击查看】

5. 查看当前软件版本

点击界面左上角菜单栏按钮【关于】>【软件版本】

6. 调整节点日志的字体大小

- <Ctrl+=> : 增大字体
- <Ctrl+-> : 减小字体

十、问题反馈

如果你在使用该软件的过程中，仍然遇到了节点配置方面的问题或有任何好的改进建议，可以通过如下途径反馈：

- 通过 e-mail 将遇到的问题截图或改进建议反馈到如下地址：wcs@xyzrobotics.ai
- 通过Teams或e-mail与我们进行联系，联系人如下：
 - junpeng.guo@xyzrobotics.ai (郭俊鹏)
 - kun.chen@xyzrobotics.ai (陈坤)