LLM Module API

目录

- LLM Module API
  - 目录
  - 概述
    - 内置功能单元
    - 使用流程
    - 通信接口
  - 数据包格式
    - 发送帧基本结构
    - 响应帧基本结构
    - 流式数据发送帧结构
    - 流式数据响应帧结构
    - 错误码
  - SYS
    - lsmode
    - hwinfo
    - reset
    - reboot
    - ping
  - AUDIO
    - setup
      - 参数说明
    - pause
    - work
    - exit
    - taskinfo
  - KWS
    - setup
      - 参数说明
      - KWS Setup
    - pause
    - work
    - exit
    - taskinfo
  - ASR
    - setup
      - 参数说明
      - ASR Setup
    - pause
    - work

        - exit
        - taskinfo
    - LLM
        - setup
            - 参数说明
            - LLM Input From ASR
            - LLM Input From UART
        - inference
            - UART inference
        - pause
        - work
        - exit
        - taskinfo
    - TTS
        - setup
            - 参数说明
            - TTS Input From LLM
            - TTS Input From UART
        - inference
            - UART inference
        - pause
        - work
        - exit
        - taskinfo
    - Application
        - Text To Speech
        - Text Assistant
        - Voice Assistant

版本	更新日期	备注
v1.0.0	2024.10.24	/

概述

LLM Module内置了KWS(唤醒词),ASR(语音识别),LLM(大语言模型),TTS(文本生成语音)等功能单元, 不同单元除了作为单独模块使用, 还能够支持配置数据工作流向进行协同, 实现更加智能的交互应用。模块支持通过UART通信方式和主机进行交互, 服务使用JSON格式数据包作为数据载体进行交互, 上手更加简单。

内置功能单元

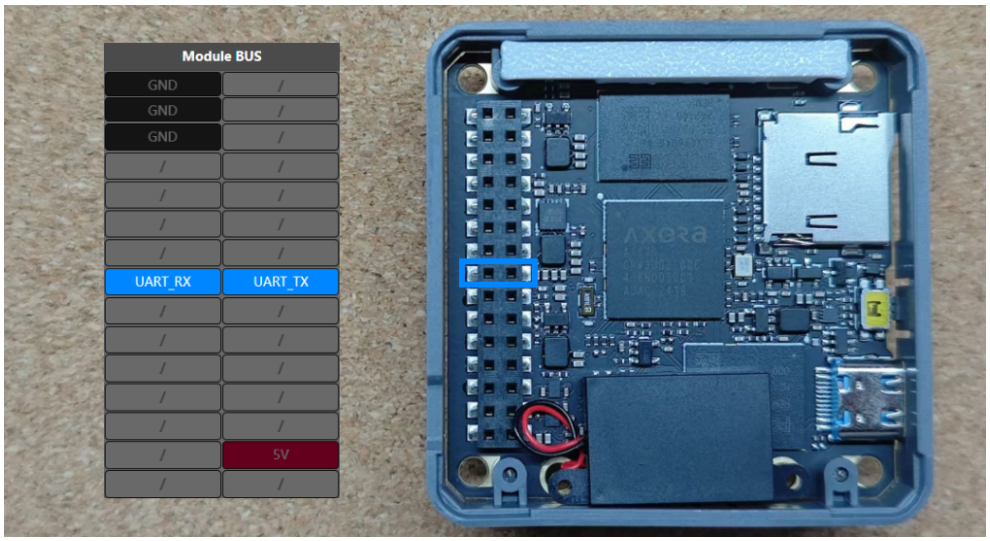
单元	单元名	单元能力
sys	系统	设置模组工作参数,获取模组运行信息
kws	语音关键词检测	检测声音中是否存在关键词
asr	语音转文本	将语音转换成文本
llm	生成式模型	根据输入的文本生成新的文本
tts	文本转语音	将文本转换成语音
audio	系统声卡	获取麦克风声音和播放声音

使用流程

- 1.将模块与M5Stack主控(Basic/M5Core2/M5Core3等)进行堆叠 / 或是直接通过USB-TTL转接板连接至TX/RX和供电, 模块等待模块绿灯亮起, 表示完成启动。
- 2.程序中初始化UART接口(引脚参数根据实际连接的设备进行配置, 接口配置为115200bps 8N1)。
- 3.参考下方使用案例, 发送初始化数据帧开启对应的单元服务。

通信接口

- LLM Module UART接口默认配置为115200bps 8N1



数据包格式

发送帧基本结构

```
{
    "request_id": "001",
    "work_id": "llm.1001",
    "action": "taskinfo",
    "object": "None",
    "data":"None"
}
```

- request_id:
  - 会话的id号,用于区分上下文,对应调用服务和响应。
- work_id:
  - 调用的服务单元时传入关键字+id, 如:llm.xxxx(id)。
  - setup初始化服务单元时,填入单元名关键字无需id, 如:llm。
- action:
  - 调用的方法,对应单元的方法, 请查看下方对应单元列表。
- object:
  - 设置传入 data 的参数结构,所有的参数结构查看参数结构列表.当无参数时可省略。
- data:
  - 传输的参数,无参数时可省略。

响应帧基本结构

```
{
  "request_id": "002",
  "work_id": "kws.1002",
  "created": 30952,
  "object": "None",
  "data":"None",
  "error":{"code":0, "message":""}
}
```

- created:
  - 完成操作的时间 Unix 时间戳,以秒为单位。
- error:
  - 状态信息,可由此字段判断服务调用失败或者成功, 更多错误码信息， 请查看下方列表。

流式数据发送帧结构

```
{
    "request_id": "4",
    "work_id": "llm.1003",
```

```
    "action": "inference",
    "object": "llm.utf-8.stream",
    "data": {
        "delta": "What's ur name?",
        "index": 0,
        "finish": true
    }
}
```

流式数据响应帧结构

```
{
    "created": 1692664605,
    "data": {
        "delta": "I'm not a person, but I'm here to help with any questions you may
have. How can I assist you today?\n",
        "finish": true,
        "index": 0
    },
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "llm.utf-8.stream",
    "request_id": "4",
    "work_id": "llm.1003"
}
```

- `index`:
  - 数据片段判断索引
- `delta`:
  - 数据片段
- `finish`:
  - true则为最后一个包

错误码

错误代码是响应中的错误代码，在error中会附带错误信息，代码主要是用于判断响应结果：

错误代码	描述	message	备注
0	操作成功！	Operation Successful!	
-1	通信信道接收状态机重置警告！	reace reset	一直发送"}"会触发此错误。用于重置json接收状态机。
-2	json 解析错误	json format error	

错误代码	描述	message	备注
-3	sys action 匹配错误	action match false	
-4	推理数据推送错误	inference data push false	
-5	模型加载失败	Model loading failed.	
-6	单元不存在	Unit Does Not Exist	
-7	未知操作	Unknown Operation	
-8	单元资源申请失败	Unit Resource Allocation Failed	
-9	单元调用失败	unit call false	
-10	模型初始化	Model init failed.	
-11	模型运行错误	Model run failed.	
-12	模块未初始化	Module has not been initialised.	
-13	模块工作中	Module already working.	
-14	模块未工作	Module is not working.	
-19	单元资源释放失败	Unit Resource Free Failed	

SYS

SYS单元用于设置模组工作参数,获取模组运行信息等。

方法	功能	输入类型	输出类型
lsmode	获取可用模型	无	sys.lsmode
hwinfo	获取cpu负载,内存负载,芯片温度	无	sys.hwinfo
reset	重启单元	无	返回重启完成json
reboot	重启系统	无	无
ping	确认系统是否可用	无	无

lsmode

- 获取可用模型

```
{
    "request_id": "001",
```

```
    "work_id": "sys",
    "action": "lsmode"
}
```

- 获取可用模型响应


```
{
    "created": 1692652687,
    "data": [
        {
            "capabilities": [
                "Automatic_Speech_Recognition"
            ],
            "input_type": [
                "sys.pcm"
            ],
            "model": "sherpa-ncnn-streaming-zipformer-zh-14M-2023-02-23",
            "output_type": [
                "asr.utf-8"
            ],
            "type": "asr"
        },
        {
            "capabilities": [
                "Automatic_Speech_Recognition"
            ],
            "input_type": [
                "sys.pcm"
            ],
            "model": "sherpa-ncnn-streaming-zipformer-20M-2023-02-17",
            "output_type": [
                "asr.utf-8"
            ],
            "type": "asr"
        },
        {
            "capabilities": [
                "Keyword_spotting"
            ],
            "input_type": [
                "sys.pcm"
            ],
            "model": "sherpa-onnx-kws-zipformer-wenetspeech-3.3M-2024-01-01",
            "output_type": [
                "kws.bool"
            ],
            "type": "kws"
        },
        {
```

```
            "capabilities": [
                "Keyword_spotting"
            ],
            "input_type": [
                "sys.pcm"
            ],
            "model": "sherpa-onnx-kws-zipformer-gigaspeech-3.3M-2024-01-01",
            "output_type": [
                "kws.bool"
            ],
            "type": "kws"
        },
        {
            "capabilities": [
                "text_generation",
                "chat"
            ],
            "input_type": "utf-8",
            "model": "qwen2.5-0.5b",
            "output_type": "utf-8",
            "type": "llm"
        },
        {
            "capabilities": [
                "Text_to_speech"
            ],
            "input_type": [
                "sys.utf-8",
                "llm.utf-8"
            ],
            "model": "single_speaker_fast",
            "output_type": [
                "tts.wav"
            ],
            "type": "tts"
        },
        {
            "capabilities": [
                "Text_to_speech"
            ],
            "input_type": [
                "sys.utf-8",
                "llm.utf-8"
            ],
            "model": "single_speaker_english_fast",
            "output_type": [
                "tts.wav"
            ],
            "type": "tts"
        }
    ],
```

```
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "sys.lsmode",
    "request_id": "001",
    "work_id": "sys"
}
```


hwinfo

- 获取cpu负载,内存负载,芯片温度


```
{
    "request_id": "001",
    "work_id": "sys",
    "action": "hwinfo"
}
```


- 获取cpu负载,内存负载,芯片温度响应(cpu_loadavg(0%), mem(18%), temperature(46°C))


```
{
    "created": 1692652642,
    "data": {
        "cpu_loadavg": 0,
        "mem": 18,
        "temperature": 46350
    },
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "sys.hwinfo",
    "request_id": "001",
    "work_id": "sys"
}
```


reset

- 系统复位指令。


```
{
    "request_id": "001",
    "work_id": "sys",
```

```
    "action": "reset"
}
```

- 开始执行系统复位。

```
{
    "created": 1692652712,
    "error": {
        "code": 0,
        "message": "llm server restarting ..."
    },
    "request_id": "001",
    "work_id": "sys"
}
```

- 完成系统复位响应。

```
{
    "request_id": "0",
    "work_id": "sys",
    "created": 1692652723,
    "error": {
        "code": 0,
        "message": "reset over"
    }
}
```

reboot

- 系统整机重启指令。

```
{
    "request_id": "001",
    "work_id": "sys",
    "action": "reboot"
}
```

- 系统整机重启指令。

```
{
    "created": 1692652822,
```

```
    "error": {
        "code": 0,
        "message": "rebooting ..."
    },
    "request_id": "001",
    "work_id": "sys"
}
```

- 注意事项: 返回消息后，系统将会重启。注意：重启时将会有一个字符串 `V0EUEURS` 被发出，字符串为系统启动时的字符串, 忽略即可。

ping

- 系统服务收发测试, 可用于模组上电后接口通信状态检查。

```
{
    "request_id": "001",
    "work_id": "sys",
    "action": "ping"
}
```

- 系统服务通信测试响应

```
{
    "created": 1692652310,
    "error": {
        "code": 0,
        "message": ""
    },
    "request_id": "001",
    "work_id": "sys"
}
```

AUDIO

AUDIO单元用于控制系统声卡, 获取麦克风声音和播放声音。提供系统音频的输入和输出。为唤醒词和语音识别单元提供系统音频输入，为文本生成语音模块提供系统音频输出。 在使用KWS和ASR功能单元前需对AUDIO单元及进行初始化。

方法	功能	输入类型	输出类型
setup	配置 audio 单元工作	audio.setup	无 (返回结果中包含成功后的work_id)
exit	结束 work_id 单元的的工作	无	无

方法	功能	输入类型	输出类型
pause	暂停任务运行	无	无
work	继续任务运行	无	无
taskinfo	获取所有的任务实例信息		audio.taskinfo

setup

- 初始化Audio单元, 配置播放音量和声卡插槽号(capcard, playcard使用默认即可)

参数说明

参数	描述	输入值
capcard	麦克风声卡的索引	系统默认声卡:0
capdevice	麦克风设备索引	板载硅麦:0
capVolume	输入的音量	0.0 ~ 10.0 (1<volume将增益, 默认值为0.5)
playcard	扬声器声卡的索引	系统默认声卡:0
playdevice	扬声器设备索引	板载扬声器:1
playVolume	输出的音量	0.0 ~ 10.0 (1<volume将增益, 默认值为0.5)

```
{
    "request_id": "1",
    "work_id": "audio",
    "action": "setup",
    "object": "audio.setup",
    "data": {
        "capcard": 0,
        "capdevice": 0,
        "capVolume": 0.5,
        "playcard": 0,
        "playdevice": 1,
        "playVolume": 0.5
    }
}
```

- 初始化Audio单元响应

```
{
    "created": 1692659008,
    "error": {
        "code": 0,
```

```
        "message": "audio setup successful"
    },
    "request_id": "1",
    "work_id": "audio.1000"
}
```

pause

- 暂停Audio单元指令

```
{
    "request_id": "1",
    "work_id": "audio.1000",
    "action": "pause"
}
```

- 暂停Audio单元指令响应

```
{
    "created": 1692659049,
    "error": {
        "code": 0,
        "message": "audio pause"
    },
    "request_id": "1",
    "work_id": "audio.1000"
}
```

work

- 开启Audio单元指令

```
{
    "request_id": "1",
    "work_id": "audio.1000",
    "action": "work",
    "object": "audio.setup",
    "data": {
        "capcard": 0,
        "capdevice": 0,
        "capVolume": 0.5,
        "playcard": 0,
        "playdevice": 1,
        "playVolume": 0.25
```

```
    }
}
```

- 开启Audio单元指令响应

```
{
    "created": 1692659297,
    "error": {
        "code": 0,
        "message": "audio work start"
    },
    "request_id": "1",
    "work_id": "audio.1000"
}
```

exit

- 结束释放Audio单元

```
{
    "request_id": "1",
    "work_id": "audio.1000",
    "action": "exit"
}
```

- 结束释放Audio单元响应

```
{
    "created": 1692659370,
    "error": {
        "code": 0,
        "message": "audio exit"
    },
    "request_id": "1",
    "work_id": "audio.1000"
}
```

taskinfo

- 查询Audio单元状态

```
// 发送数据
{
    "request_id": "1",
    "work_id": "audio.1000",
    "action": "taskinfo"
}
```

- Audio单元运行中响应

```
{
    "created": 1692659454,
    "data": "running",
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "audio.state",
    "request_id": "1",
    "work_id": "audio.1000"
}
```

- Audio单元已停止响应

```
{
    "created": 1692659499,
    "data": "stopped",
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "audio.state",
    "request_id": "1",
    "work_id": "audio.1000"
}
```

- Audio单元已释放响应

```
{
    "created": 1692659403,
    "data": "deinit",
    "error": {
        "code": 0,
        "message": ""
```

```
    },
    "object": "audio.state",
    "request_id": "1",
    "work_id": "audio.1000"
}
```

KWS

KWS单元用于唤醒关键词检测。

方法	功能	输入类型	输出类型
setup	配置 kws 单元工作	kws.setup	无 (返回结果中包含成功后的work_id)
pause	暂停任务运行	无	无
work	继续任务运行	无	无
exit	结束 work_id 单元的的工作	无	无
taskinfo	获取所有的任务实例信息		kws.taskinfo

setup

- 初始化KWS单元, 并配置为中文/英文识别model。(注意: kws唤醒词字段不允许中文/英文混合)

参数说明

参数	描述	输入值
model	转换模型	英文模型: "sherpa-onnx-kws-zipformer-gigaspeech-3.3M-2024-01-01" 中文模型: "sherpa-onnx-kws-zipformer-wenetspeech-3.3M-2024-01-01"
kws	KWS唤醒词文本设置	不允许中文/英文混合, 英文要求全大写
enoutput	启用UART输出	启用: true 禁用: false

KWS Setup

- 初始化KWS单元, 并配置为英文识别model。

```
{
    "request_id": "2",
    "work_id": "kws",
    "action": "setup",
    "object": "kws.setup",
    "data": {
        "model": "sherpa-onnx-kws-zipformer-gigaspeech-3.3M-2024-01-01",
```

```
        "response_format": "kws.bool",
        "input": "sys.pcm",
        "enoutput": true,
        "kws": "HELLO"
    }
}
```

- 初始化KWS响应(注意：setup过程需要耗费约9s时间)

```
{
    "created": 1692660576,
    "error": {
        "code": 0,
        "message": "kws setup successful"
    },
    "request_id": "2",
    "work_id": "kws.1001"
}
```

- KWS唤醒词触发后响应

```
{
    "created": 1692660576,
    "error": {
        "code": 0,
        "message": "kws setup successful"
    },
    "request_id": "2",
    "work_id": "kws.1001"
}
```

pause

- 暂停KWS单元指令

```
{
    "request_id": "2",
    "work_id": "kws.1001",
    "action": "pause"
}
```

- 暂停Audio单元指令响应

```
{
    "created": 1692660626,
    "error": {
        "code": 0,
        "message": "kws pause"
    },

    "request_id": "2",
    "work_id": "kws.1001"
}
```


work

- 开启KWS单元指令


```
{
    "request_id": "2",
    "work_id": "kws.1001",
    "action": "work"
}
```


- 开启KWS单元指令响应


```
{
    "created": 1692660651,
    "error": {
        "code": 0,
        "message": "kws work"
    },
    "request_id": "2",
    "work_id": "kws.1001"
}
```


exit

- 结束释放KWS单元


```
{
  "request_id": "2",
  "work_id": "kws.1001",
  "action": "exit"
}
```

- 结束释放KWS单元响应

```
{
  "created": 1692654383,
  "error": {
    "code": 0,
    "message": "kws exit"
  },
  "request_id": "2",
  "work_id": "kws.1001"
}
```

taskinfo

- 查询KWS单元状态

```
{
  "created": 1692654383,
  "error": {
    "code": 0,
    "message": "kws exit"
  },
  "request_id": "2",
  "work_id": "kws.1001"
}
```

- KWS单元运行中响应

```
{
  "created": 1692654305,
  "error": {
    "code": 0,
    "message": ""
  },
  "object": "kws.state",
  "data":"runing",
  "request_id": "2",
  "work_id": "kws.1001"
}
```

- KWS单元已停止响应

```
{
  "created": 1692654535,
  "error": {
    "code": 0,
    "message": ""
  },
  "object": "kws.state",
  "data":"stop",
  "request_id": "2",
  "work_id": "kws.1001"
}
```


- KWS单元已释放响应


```
{
  "created": 1692654452,
  "error": {
    "code": 0,
    "message": ""
  },
  "object": "kws.state",
  "data":"deinit",
  "request_id": "2",
  "work_id": "kws.0"
}
```


ASR

ASR单元用于将语音转换成文本。

方法	功能	输入类型	输出类型
setup	配置 asr 单元工作	asr.setup	无 (返回结果中包含成功后的work_id)
pause	暂停任务运行	无	无
work	继续任务运行	无	无
exit	结束 work_id 单元的的工作	无	无
taskinfo	获取所有的任务实例信息		asr.taskinfo

setup

- 初始化ASR单元, 并配置为中文/英文转换模型。

参数说明

参数	描述	输入值
model	转换模型	英文模型: "sherpa-ncnn-streaming-zipformer-20M-2023-02-17" 中文模型: "sherpa-ncnn-streaming-zipformer-zh-14M-2023-02-23"
response_format	输出格式	普通输出: "asr.utf-8" 流式输出: "asr.utf-8.stream"
input	输入	LLM输入: "llm.xxx"(输入llm单元的work_id) UART输入: "tts.utf-8" UART流式输入: "tts.utf-8.stream"
enkws	是否支持通过KWS唤醒	可通过KWS唤醒, 并进行ASR: true 不通过KWS唤醒, ASR单元将持续工作: false
rule1	唤醒到未识别到内容超时时间	单位:秒
rule2	识别最大间隔时间	单位:秒
rule3	识别最长超时时间	单位:秒
enoutput	启用UART输出	启用: true 禁用: false

ASR Setup

- 初始化ASR单元, 并配置为英文语音转换model。

```
{
    "request_id": "3",
    "work_id": "asr",
    "action": "setup",
    "object": "asr.setup",
    "data": {
        "model": "sherpa-ncnn-streaming-zipformer-20M-2023-02-17",
        "response_format": "asr.utf-8",
        "input": "sys.pcm",
        "enoutput": true,
        "enkws":true,
        "rule1":2.4,
        "rule2":1.2,
        "rule3":30
    }
}
```

- 初始化ASR响应

```
{
    "created": 1692667736,
    "error": {
        "code": 0,
        "message": "asr setup successful"
    },
    "request_id": "3",
    "work_id": "asr.1002"
}
```


- ASR触发后响应


```
{
    "created": 1692655176,
    "data": {
        "delta": " hello",
        "index": "0"
    },
    "object": "asr.stream",
    "request_id": "004",
    "work_id": "asr.1003"
}
```


pause

- 暂停ASR单元指令


```
{
    "request_id": "3",
    "work_id": "asr.1002",
    "action": "pause"
}
```


- 暂停ASR单元指令响应


```
{
    "created": 1692670174,
    "error": {
        "code": 0,
        "message": "asr pause"
    },
    "request_id": "3",
```

```
    "work_id": "asr.1002"
}
```

work

- 开启ASR单元指令

```
{
    "request_id": "3",
    "work_id": "asr.1002",
    "action": "pause"
}
```

- 开启ASR单元指令响应

```
{
    "created": 1692670213,
    "error": {
        "code": 0,
        "message": "asr work"
    },
    "request_id": "3",
    "work_id": "asr.1002"
}
```

exit

- 结束释放ASR单元

```
{
    "request_id": "3",
    "work_id": "asr.1002",
    "action": "exit"
}
```

- 结束释放ASR单元响应

```
{
    "created": 1692670254,
    "error": {
        "code": 0,
        "message": "asr exit"
```

```
    },
    "request_id": "3",
    "work_id": "asr.1002"
}
```

taskinfo

- 查询ASR单元状态

```
{
    "request_id": "3",
    "work_id": "asr.1002",
    "action": "taskinfo"
}
```

- ASR单元运行中响应

```
{
    "created": 1692669923,
    "data": "running",
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "asr.state",
    "request_id": "3",
    "work_id": "asr.1002"
}
```

- ASR单元已停止响应

```
{
  "created": 1692653792,
  "data": "stopped",
  "error": {
    "code": 0,
    "message": ""
  },
  "object": "asr.state",
  "request_id": "3",
  "work_id": "asr.1002"
}
```

- ASR单元已释放响应

```
{
    "created": 1692669874,
    "data": "deinit",
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "asr.state",
    "request_id": "3",
    "work_id": "asr.0"
}
```

LLM

LLM大语言模型单元, 能够根据输入的文本生成新的文本回复。

方法	功能	输入类型	输出类型
setup	配置 llm 单元工作	llm.setup	无 (返回结果中包含成功后的work_id)
inference	推理数据	典型 llm.utf-8 (模型差异可由 sys.lsmode 得到)	无 (只返回数据发送结果,推理完成后会更去配置决定是否输出推理结果)
pause	暂停任务运行	无	无
work	继续任务运行	无	无
exit	结束 work_id 单元的的工作	无	无
taskinfo	获取所有的任务实例信息		llm.taskinfo

setup

- 初始化LLM单元, 并配置指定模型. 目前出厂预置模型:
  - `qwen2.5-0.5b`

参数说明

参数	描述	输入值
model	转换模型	预置模型 "qwen2.5-0.5b"
response_format	输出格式	普通输出: "llm.utf-8" 流式输出: "llm.utf-8.stream"

参数	描述	输入值
input	输入	ASR输入: "asr.xxx"(输入asr单元的work_id) UART输入: "llm.utf-8" UART流式输入: "llm.utf-8.stream"
enkws	KWS唤醒是否终止过程	KWS打断过程: true KWS不打断过程: false
max_length	配置最大输出token(最大返回推理文本长度)	最大值: 1024, 推荐使用127
prompt	模型初始化提示词	
enoutput	启用UART输出	启用: true 禁用: false

LLM Input From ASR

- 初始化LLM单元, 并配置ASR(语音转文本)作为输入数据

```
// Input from ASR
{
    "request_id": "4",
    "work_id": "llm",
    "action": "setup",
    "object": "llm.setup",
    "data": {
        "model": "qwen2.5-0.5b",
        "response_format": "llm.utf-8.stream",
        "input": "asr.1001",
        "enoutput": true,
        "enkws": true,
        "max_token_len": 127,
        "prompt": "You are a knowledgeable assistant capable of answering various
questions and providing information."
    }
}
```

LLM Input From UART

- 初始化LLM单元, 并配置UART接口作为输入数据

```
// Input from UART
{
    "request_id": "4",
    "work_id": "llm",
    "action": "setup",
```

```
    "object": "llm.setup",
    "data": {
        "model": "qwen2.5-0.5b",
        "response_format": "llm.utf-8",
        "input": "llm.utf-8.stream",
        "enoutput": true,
        "enkws": true,
        "max_token_len": 127,
        "prompt": "You are a knowledgeable assistant capable of answering various
questions and providing information."
    }
}
```

- 初始化LLM单元响应

```
{
    "created": 1692664107,
    "data": "None",
    "error": {
        "code": 0,
        "message": "llm setup successful"
    },
    "object": "None",
    "request_id": "4",
    "work_id": "llm.1003"
}
```

inference

UART inference

- 通过过UART提交推理数据内容

```
// 流式发送数据 Streaming Input
{
    "request_id": "4",
    "work_id": "llm.1003",
    "action": "inference",
    "object": "llm.utf-8.stream",
    "data": {
        "delta": "What's ur name?",
        "index": 0,
        "finish": true
    }
}
// 发送数据 Non-Streaming Input
```

```
{
    "request_id": "4",
    "work_id": "llm.1003",
    "action": "inference",
    "object": "llm.utf-8",
    "data": "What's ur name?"
}
```

- 推理响应数据。

```
{
    "created": 1692664605,
    "data": {
        "delta": "I'm not a person, but I'm here to help with any questions you may
have. How can I assist you today?\n",
        "finish": true,
        "index": 0
    },
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "llm.utf-8.stream",
    "request_id": "4",
    "work_id": "llm.1003"
}
```

pause

- 暂停LLM单元指令

```
{
    "request_id": "4",
    "work_id": "llm.1003",
    "action": "pause"
}
```

- 暂停LLM单元指令响应

```
{
    "created": 1692664941,
    "error": {
        "code": 0,
        "message": "llm pause"
    },
```

```
    "request_id": "4",
    "work_id": "llm.1003"
}
```

work

- 开启LLM单元指令

```
{
    "request_id": "4",
    "work_id": "llm.1003",
    "action": "work"
}
```

- 开启LLM单元指令响应

```
{
    "created": 1692664972,
    "error": {
        "code": 0,
        "message": "llm work"
    },
    "request_id": "4",
    "work_id": "llm.1003"
}
```

exit

- 结束释放LLM单元

```
{
    "request_id": "4",
    "work_id": "llm.1003",
    "action": "exit"
}
```

- 结束释放LLM单元响应

```
{
    "created": 1692664858,
    "data": "None",
    "error": {
```

```
        "code": 0,
        "message": "llm exit"
    },
    "object": "None",
    "request_id": "4",
    "work_id": "llm.1003"
}
```

taskinfo

- 查询LLM单元状态

```
{
    "request_id": "4",
    "work_id": "llm.1003",
    "action": "taskinfo"
}
```

- LLM单元运行中响应

```
{
    "created": 1692664730,
    "data": "running",
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "llm.state",
    "request_id": "4",
    "work_id": "llm.1003"
}
```

- LLM单元已停止响应

```
{
    "created": 1692664823,
    "data": "stopped",
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "llm.state",
    "request_id": "4",
```

```
    "work_id": "llm.1003"
}
```

- LLM单元已释放响应

```
{
    "created": 1692664881,
    "data": "deinit",
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "llm.state",
    "request_id": "4",
    "work_id": "llm.1003"
}
```

TTS

TTS单元用于将文本转换成语音。

方法	功能	输入类型	输出类型
setup	配置 tts 单元工作	tts.setup	无 (返回结果中包含成功后的work_id)
inference	推理数据	典型 tts.utf-8 (模型差异可由 sys.lsmode 得到)	无 (只返回数据发送结果,推理完成后会更去配置决定是否输出推理结果)
pause	暂停任务运行	无	无
work	继续任务运行	无	无
exit	结束 work_id 单元的的工作	无	无
taskinfo	获取所有的任务实例信息		tts.taskinfo

setup

- 初始化TTS单元, 并配置为中文/英文转换模型。

参数说明

参数	描述	输入值
model	转换模型	英文模型: "single_speaker_english_fast" 中文模型: "single_speaker_fast"
input	输入	LLM输入: "llm.xxx"(输入llm单元的work_id) UART输入: "tts.utf-8" UART流式输入: "tts.utf-8.stream"
enkws	KWS唤醒是否终止过程	KWS打断过程: true KWS不打断过程: false
enoutput	启用UART输出	启用: true 禁用: false

TTS Input From LLM

- 初始化TTS单元, 并配置为英文文本转换model, 转换文本输入来源配置为LLM推理结果。

```
// Input from LLM
{
    "request_id": "5",
    "work_id": "tts",
    "action": "setup",
    "object": "tts.setup",
    "data": {
        "model": "single_speaker_english_fast",
        "response_format": "tts.base64.wav",
        "input": "llm.1004",
        "enoutput": true,
        "enkws": true
    }
}
```

TTS Input From UART

- 初始化TTS单元, 并配置为英文文本转换model, 转换文本输入来源配置UART指令流式输入。

```
// Input from UART
{
  "request_id": "5",
  "work_id": "tts",
  "action": "setup",
  "object": "tts.setup",
  "data": {
        "model": "single_speaker_english_fast",
        "response_format": "tts.base64.wav",
```

```
        "input": "tts.utf-8.stream",
        "enoutput": true,
        "enkws": true
    }
}
```

- TTS单元初始化响应

```
{
    "created": 1692668824,
    "error": {
        "code": 0,
        "message": "tts setup successful"
    },
    "request_id": "5",
    "work_id": "tts.1004"
}
```

inference

UART inference

- 通过UART提交TTS转换数据内容。一种模型同时仅支持一种语言，转换不同语言时请使用`exit`释放TTS单元后重新setup。
- 注意事项: 转换文本要求以句号结尾:
    - 使用英文文本时, 要求英文结尾句号`.`(半角符号)
    - 使用中文文本时, 要求中文结尾句号`。`(全角符号)
    - 句子分隔符使用`,`(半角符号)

```
// 流式发送数据 Streaming Input
{
    "request_id": "4",
    "work_id": "tts.1004",
    "action": "inference",
    "object": "tts.utf-8.stream",
    "data": {
        "delta":"I don't know what your name.",
        "index":0,
        "finish":true
    }
}

// 发送数据 Non-Streaming Input
{
```

```
    "request_id": "4",
    "work_id": "tts.1004",
    "action": "inference",
    "object": "tts.utf-8",
    "data": "I don't know what your name."
}
```

pause

- 暂停TTS单元指令

```
{
    "request_id": "5",
    "work_id": "tts.1004",
    "action": "pause"
}
```

- 暂停TTS单元指令响应

```
{
    "created": 1692668916,
    "error": {
        "code": 0,
        "message": "tts pause"
    },
    "request_id": "5",
    "work_id": "tts.1004"
}
```

work

- 开启TTS单元指令

```
{
    "request_id": "5",
    "work_id": "tts.1004",
    "action": "work"
}
```

- 开启TTS单元指令响应

```
{
    "created": 1692668944,
    "error": {
        "code": 0,
        "message": "tts work"
    },
    "request_id": "5",
    "work_id": "tts.1004"
}
```

exit

- 结束释放TTS单元

```
{
    "request_id": "5",
    "work_id": "tts.1004",
    "action": "exit"
}
```

- 结束释放TTS单元响应

```
{
    "created": 1692669052,
    "error": {
        "code": 0,
        "message": "tts exit"
    },
    "request_id": "5",
    "work_id": "tts.1004"
}
```

taskinfo

- 查询TTS单元状态

```
{
    "request_id": "5",
    "work_id": "tts.1004",
    "action": "taskinfo"
}
```

- TTS单元运行中响应


```
{
    "created": 1692668878,
    "data": "running",
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "tts.state",
    "request_id": "5",
    "work_id": "tts.1004"
}
```


- TTS单元已停止响应


```
{
    "created": 1692668968,
    "data": "stopped",
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "tts.state",
    "request_id": "5",
    "work_id": "tts.1004"
}
```


- TTS单元已释放响应


```
{
    "created": 1692669081,
    "data": "deinit",
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "tts.state",
    "request_id": "5",
    "work_id": "tts.1004"
}
```


Application

Text To Speech

通过TTS单元实现文本转换语音播放。 (TTS)

- 1.初始化Audio单元

```
{
    "request_id": "1",
    "work_id": "audio",
    "action": "setup",
    "object": "audio.setup",
    "data": {
        "capcard": 0,
        "capdevice": 0,
        "capVolume": 0.5,
        "playcard": 0,
        "playdevice": 1,
        "playVolume": 0.5
    }
}
```

- 初始化Audio单元响应

```
{
    "created": 1692652475,
    "error": {
        "code": 0,
        "message": "audio setup successful"
    },
    "request_id": "1",
    "work_id": "audio.1000"
}
```

- 2.初始化TTS单元, 并配置为英文文本转换model, 转换文本输入来源配置UART指令输入。

```
// Input from UART
{
  "request_id": "5",
  "work_id": "tts",
  "action": "setup",
  "object": "tts.setup",
  "data": {
        "model": "single_speaker_english_fast",
        "response_format": "tts.base64.wav",
        "input": "tts.utf-8",
        "enoutput": true,
```

```
        "enkws": true
    }
}
```

- TTS单元初始化响应

```
{
    "created": 1692652569,
    "error": {
        "code": 0,
        "message": "tts setup successful"
    },
    "request_id": "5",
    "work_id": "tts.1001"
}
```

- 3.输入文本，开始TTS转换。

```
{
    "request_id": "4",
    "work_id": "tts.1001",
    "action": "inference",
    "object": "tts.utf-8",
    "data": "Hello My Friend."
}
```

Text Assistant

通过文本方式输入内容至LLM模型, 完成推理后以语音形式播放。 (LLM+TTS)

- 1.初始化Audio单元

```
{
    "request_id": "1",
    "work_id": "audio",
    "action": "setup",
    "object": "audio.setup",
    "data": {
        "capcard": 0,
        "capdevice": 0,
        "capVolume": 0.5,
        "playcard": 0,
        "playdevice": 1,
        "playVolume": 0.5
```

```
    }
}
```

- 初始化Audio单元响应

```
{
    "created": 1692652330,
    "error": {
        "code": 0,
        "message": "audio setup successful"
    },
    "request_id": "1",
    "work_id": "audio.1000"
}
```

- 2.初始化LLM单元, 并配置UART接口作为输入数据

```
// Input from UART
{
    "request_id": "4",
    "work_id": "llm",
    "action": "setup",
    "object": "llm.setup",
    "data": {
        "model": "qwen2.5-0.5b",
        "response_format": "llm.utf-8",
        "input": "llm.utf-8",
        "enoutput": true,
        "enkws": true,
        "max_token_len": 127,
        "prompt": "You are a knowledgeable assistant capable of answering various
questions and providing information."
    }
}
```

- 初始化LLM单元响应

```
{
    "created": 1692652323,
    "error": {
        "code": 0,
        "message": "llm setup successful"
    },
    "request_id": "4",
```

```
    "work_id": "llm.1001"
}
```

- 3.初始化TTS单元, 并配置为英文文本转换model, 转换文本输入来源配置为LLM推理结果。

```
// Input from LLM
{
    "request_id": "5",
    "work_id": "tts",
    "action": "setup",
    "object": "tts.setup",
    "data": {
        "model": "single_speaker_english_fast",
        "response_format": "tts.base64.wav",
        "input": "llm.1001",
        "enoutput": true,
        "enkws": true
    }
}
```

- 初始化TTS单元响应

```
{
    "created": 1692652354,
    "error": {
        "code": 0,
        "message": "tts setup successful"
    },
    "request_id": "5",
    "work_id": "tts.1002"
}
```

- 4.通过过UART提交推理数据内容

```
// 发送数据 Non-Streaming Input
{
    "request_id": "4",
    "work_id": "llm.1001",
    "action": "inference",
    "object": "llm.utf-8",
    "data": "What's ur name?"
}
```

- 5.推理响应数据, 同时输出播放语音。

```
{
    "created": 1692652407,
    "data": "I'm not a person, but I'm here to help with any questions you may have.
How can I assist you today?\n",
    "error": {
        "code": 0,
        "message": ""
    },
    "object": "llm.utf-8",
    "request_id": "4",
    "work_id": "llm.1001"
}
```


Voice Assistant

通过KWS实现唤醒->触发ASR实现语音转换文本->将其转换内容作为LLM输入用作推理->最后将推理输出结果通过TTS输出语音。 (KWS+ASR+LLM+TTS)

- 1.初始化Audio单元


```
{
    "request_id": "1",
    "work_id": "audio",
    "action": "setup",
    "object": "audio.setup",
    "data": {
        "capcard": 0,
        "capdevice": 0,
        "capVolume": 0.5,
        "playcard": 0,
        "playdevice": 1,
        "playVolume": 0.5
    }
}
```


- 初始化Audio单元响应


```
{
    "created": 1692652330,
    "error": {
        "code": 0,
        "message": "audio setup successful"
    },
    "request_id": "1",
```

```
    "work_id": "audio.1000"
}
```

- 2.初始化KWS单元, 并配置为英文识别model, 唤醒词为"HELLO"。

```
{
    "request_id": "2",
    "work_id": "kws",
    "action": "setup",
    "object": "kws.setup",
    "data": {
        "model": "sherpa-onnx-kws-zipformer-gigaspeech-3.3M-2024-01-01",
        "response_format": "kws.bool",
        "input": "sys.pcm",
        "enoutput": true,
        "kws": "HELLO"
    }
}
```

- 初始化KWS响应(注意：setup过程需要耗费约9s时间)

```
{
    "created": 1692652559,
    "error": {
        "code": 0,
        "message": "kws setup successful"
    },
    "request_id": "2",
    "work_id": "kws.1001"
}
```

- 3.初始化ASR单元, 并配置为英文语音转换model, 并设置KWS触发ASR。

```
{
    "request_id": "3",
    "work_id": "asr",
    "action": "setup",
    "object": "asr.setup",
    "data": {
        "model": "sherpa-ncnn-streaming-zipformer-20M-2023-02-17",
        "response_format": "asr.utf-8",
        "input": "sys.pcm",
        "enoutput": true,
        "enkws":true,
        "rule1":2.4,
```

```
        "rule2":1.2,
        "rule3":30
    }
}
```

- 初始化ASR响应

```
{
    "created": 1692652705,
    "error": {
        "code": 0,
        "message": "asr setup successful"
    },
    "request_id": "3",
    "work_id": "asr.1002"
}
```

- 4.初始化LLM单元, 并配置ASR(语音转文本)作为输入数据

```
// Input from ASR
{
    "request_id": "4",
    "work_id": "llm",
    "action": "setup",
    "object": "llm.setup",
    "data": {
        "model": "qwen2.5-0.5b",
        "response_format": "llm.utf-8.stream",
        "input": "asr.1002",
        "enoutput": true,
        "enkws": true,
        "max_token_len": 127,
        "prompt": "You are a knowledgeable assistant capable of answering various
questions and providing information."
    }
}
```

- 初始化LLM响应

```
{
    "created": 1692653061,
    "error": {
        "code": 0,
        "message": "llm setup successful"
    },
```

```
    "request_id": "4",
    "work_id": "llm.1003"
}
```

- 5.初始化TTS单元, 并配置为英文文本转换model, 转换文本输入来源配置为LLM推理结果。

```
// Input from LLM
{
    "request_id": "5",
    "work_id": "tts",
    "action": "setup",
    "object": "tts.setup",
    "data": {
        "model": "single_speaker_english_fast",
        "response_format": "tts.base64.wav",
        "input": "llm.1003",
        "enoutput": true,
        "enkws": true
    }
}
```

- 初始化TTS单元响应

```
{
    "created": 1692653109,
    "error": {
        "code": 0,
        "message": "tts setup successful"
    },
    "request_id": "5",
    "work_id": "tts.1004"
}
```

- 6.通过关键字"HELLO"唤醒, 然后输入语音交互。