

广东新岸线计算机系统芯片有限公司

Guangdong Nufont CSC Co., Ltd

NL6621-NuAgent SDK

设计概念

林辉

2015 年 07 月 03 日

Change Log

Date	Version	Types (New/Delete/ Modify)	Editor	Description
2015-07-03	0.01.01	New	林辉	完成文档基本框架
2015-12-21	0.02.01	New,Modify	张汇楼	添加 BSP、模拟串口接收发送、Sniffer 配置模式

目录

目录.....	2
1. 引言.....	4
1.1 概述.....	4
2. NuAgent SDK 基础信息.....	5
2.1 固件产品信息.....	5
2.2 固件系统资源.....	5
3. NuAgent SDK 软件框架.....	6
3.1 NuAgent SDK 目录结构.....	6
3.2 SDK 软件框架流程图.....	7
4. NuAgent SDK 编程说明.....	8
4.1 NuAgent 日志系统.....	8
4.2 LED 指示与系统软复位.....	8
4.3 配置模式.....	9
4.3.1 SoftAP.....	9
4.3.2 DirectConfig.....	9
4.4 Uart 收发数据.....	10
4.4.1 串口收发机制.....	10
4.5 模拟串口功能说明.....	10
4.6 Sniffer 一键配置流程.....	10
4.7 BSP 外设接口 Demo 测试.....	10
4.7.1 GPIO 设置输出输入模式.....	11
4.7.2 GPIO 中断模式.....	11
4.7.3 定时器模式.....	12
4.7.4 看门狗.....	12
4.7.5 I2C 读写 EEPROM.....	12
4.7.6 QSPI 读写 FLASH.....	12
4.7.7 SPI 传输.....	12

4.7.8 I2S 读写数据.....	12
4.7.9 SDIO 读写数据.....	13
4.7.10 DMA 内存搬运.....	13
5. NuAgent SDK 获取.....	14
6. 注意事项.....	15
6.1 库的使用.....	15

1. 引言

1.1 概述

本文描述 NL6621 NuAgent SDK 的使用说明。对接入企业云（私有云以及互联网云）提供基本的设备端开发框架。以方便已有私有云的企业能够快速的使用 NL6621 芯片接入到企业云。

由于不同的客户具体的设备实现功能不一样，相应的代码实现也不一样（例如智能插座和智能灯的实现，智能灯挂载天花，不需要 DirectConfig 触发，可以不实现 DirectConfig 功能，只需要实现 softap 配置功能即可）。本文档基于 NuAgent SDK 只作为参考范例，提供基本的 NL6621 开发代码框架，描述设备的联网/建网、创建 UDP/TCP 服务器端、创建 TCP Client 客户端。

客户根据自己的需求可以对相应功能的裁剪和实现。这里描述的接口只为了客户能够更好的理解 NL6621 的一些资源和接口的使用。本文档以“NL6621 SDK 用户使用手册.pdf”为基准，添加通用编程接口。相关接口和代码的变动以“NL6621 SDK 用户使用手册.pdf”为准。

2. NuAgent SDK 基础信息

2.1 固件产品信息

软件版本号: V020000

Firmware 版本号: 15121017

Firmware 版本号用于描述 SDK 软件发布的时间，共 8 字节的用日期和时间来描述；格式为：年（2 位）月（2 位）日（2 位）时（2 位 24 小时格式），比如：14112521，表示 2014 年 11 月 25 日 21 点发布的版本。

固件发布名称: NuAgent_V020000_15122109_RAMDBG.bin

格式为: NuAgent+软件版本号+Firmware 版本号+Ram/Rom 版本。

注：详细产品固件信息设置位于 ../Project/PrjSdkRam/bin.bat 文件中，该文件会在 Keil 编译后调用，用于生成固件并将固件拷贝到 ./Tool 目录下（在标准 SDK 中，不会讲生成的固件拷贝到 Tool 目录中）。

2.2 固件系统资源

NuAgent 基本协议栈以及系统代码的 bin 文件大小及 Data Sram:

ROM 工程 (Code/Sram) : 128KBytes /179KBytes

RAM 工程 (Code/Sram) : 163KBytes/179KBytes

留给客户的内存空间为:

ROM 工程 (Code/Sram) : 64KBytes/(77+20)KBytes

RAM 工程 (Code/Sram) : 29KBytes/(77+20)KBytes

注：工程默认采用 O3 优化级别，没有调用的代码是不会占空间，Data Sram 内存动态分配空间默认是 20KB，需要增大内存，宏定义 OS_DMEN_POOL_SIZE 修改大小，内存使用强烈建议使用 OSMMalloc，OSMFree 接口。

3. NuAgent SDK 软件框架

3.1 NuAgent SDK 目录结构

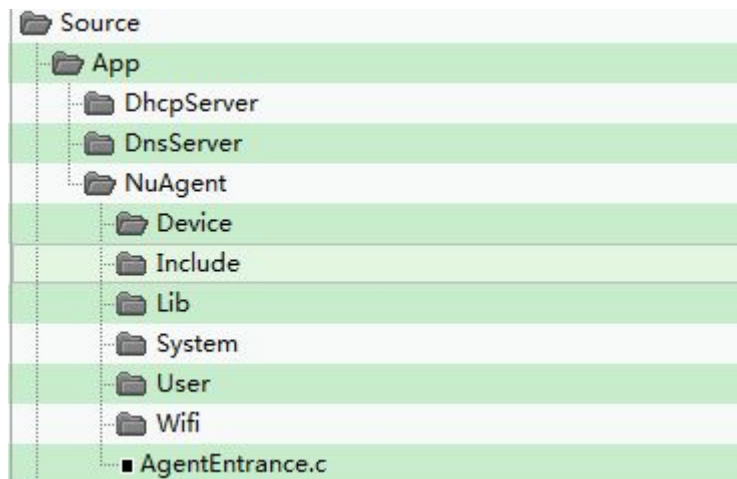


Figure 1 NuAgent SDK 工程目录

NL6621 云端接入工程目录“NuAgent”如上图所示：

- ◆ Device 目录存放 NL6621 外设接口的使用范例；
- ◆ Include 目录存放系统主要头文件；
- ◆ Lib 目录存放 SDK 的扩展库；
- ◆ System 目录存放 SDK 的核心处理文件；
- ◆ User 目录存放用户与云端接入的代码文件；
- ◆ Wifi 目录存放 NL6621 WIFI 相关的操作接口目录；

注：企业云用户在移植相关代码到 NL6621 上时，根据以上的代码存放框架，主要将用户的私有代码存放至 User 目录。

3.2 SDK 软件框架流程图

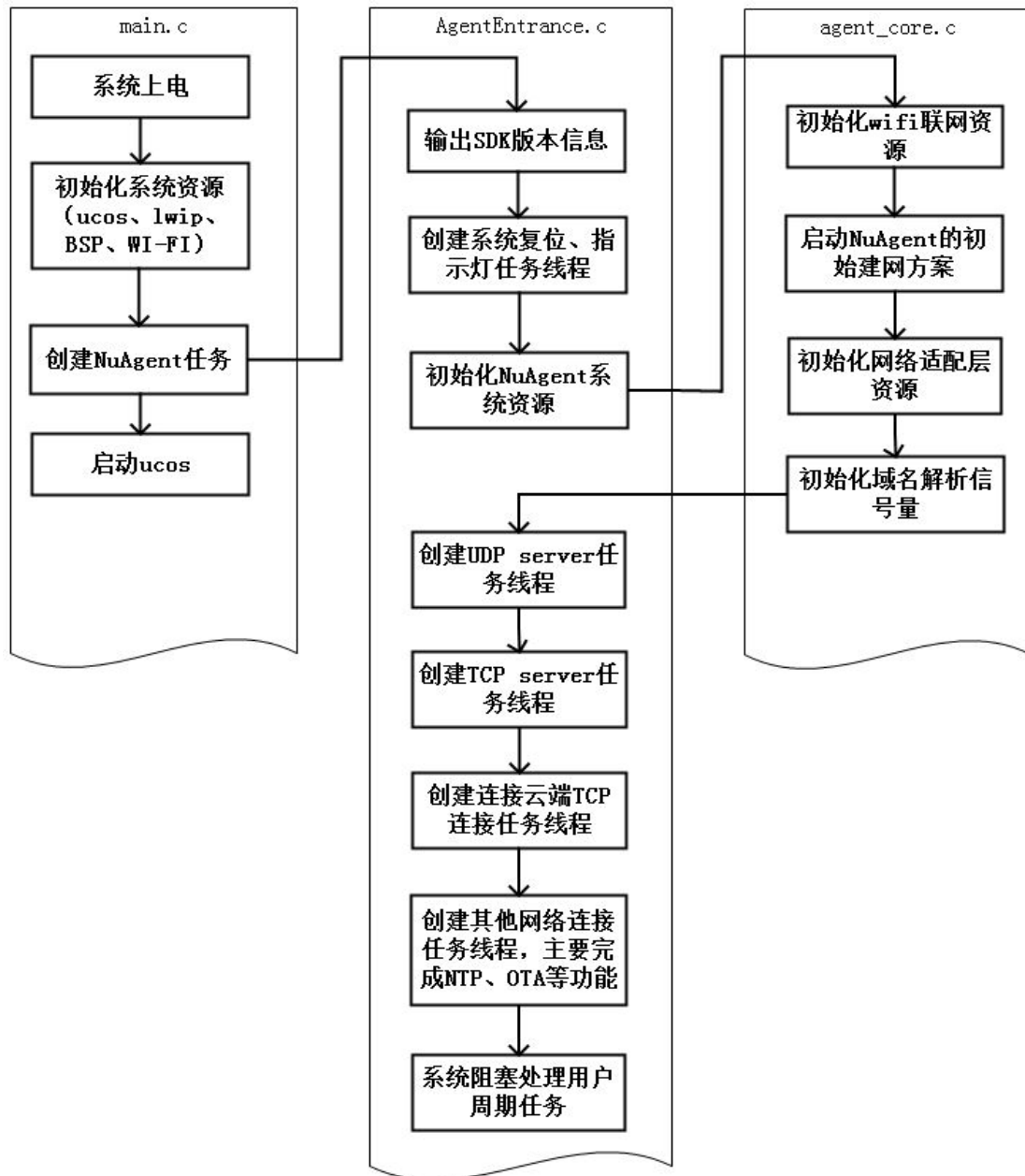


Figure 2 NuAgent 系统启动流程图

从上图可知 NuAgent 系统的入口为 main.c，系统主任务进程为 AgentEntrance.c 文件，系统各个任务的执行流程以及初始化内容在 agent_core.c 文件。

4. NuAgent SDK 编程说明

4.1 NuAgent 日志系统

为了减少固件占用 code Sram 中的空间。其中系统的 debug 信息占用了大量的代码空间，因此这里定义了一套新的日志系统。

相关代码位于 Source/App/NuAgent/include/log.h 文件。自定义 5 个级别的日志信息输出，分别为 ERROR,WARNING,NOTICE,INFO,DEBUG。使用宏 DEBUG_LEVEL_SWITCH 作为开关，开启相关级别的打印信息。该级别信息在编译的时候决定相关级别的 debug 信息是否编进固件。

日志系统的各个级别信息使用规范：

log_err：系统错误打印

log_warn：系统警告打印

log_notice：系统运行必要信息打印

log_info：系统运行状态信息打印

log_debug：系统调试信息打印

以上 5 个级别的打印中，log_err、log_warn 和 log_notice 级别信息默认打开，info 和 debug 级别信息默认关闭（DEBUG_LEVEL_SWITCH 值为 0x07）。

注：在 log.h 文件 REAL_UART_USED 宏定义打开，log 默认使用硬件串口输出（即 GPIO12），如果 REAL_UART_USED 宏定义屏蔽，log 默认使用模拟串口输出（即 GPIO5）在开发正式的产品时，最终的产品应该屏蔽无用的打印信息，只保留 log_err 级别信息即可，如果有使用串口透传，那么相关打印都需要屏蔽掉。

4.2 LED 指示与系统软复位

NL6621 需要两个 GPIO 作为系统基本外设：一个 LED 指示灯以及一个触发按键：

LED 指示灯（GPIO 9）：作为 LED 指示灯所用

- 1) 模组处于 SoftAP：指示灯灭 3 秒，亮 1 秒
- 2) 模组处于 DirectConfig：指示灯以 200 毫秒快闪
- 3) 模组连接 AP 成功，但没有连接云端：指示灯常亮 1 秒，灭 1 秒
- 4) 模组成功接入云端：指示灯常亮
- 5) WIFI 停止工作：指示灯常亮

触发按键（GPIO 10）：

系统复位以及 DirectConfig 模式切换，其中系统上电启动阶段按下该按键系统进入 DirectConfig 模式。如果没有触发 DirectConfig 模式，那么初始化系统 reset 任务，用户只要按住复位键超过 4 秒则系统重启（代码：Agent_network_init()接口）。

注：系统软复位功能，会将保持在 norflash 中的用户信息擦除，在系统重启后，会默认进入 SoftAP 模式。针对 NF-210S 和 NF-210D 两款模组，没有接出芯片的硬件复位管脚，则

使用软件进行复位，如果客户只需要复位，并且不需要擦除保存的网络连接信息，则客户可以考虑使用硬件复位功能。

4.3 配置模式

NL6621 支持 2 种配置方式，分别是 SoftAP、Directconfig、Sniffer 模式配置（此模式只提供配置流程参考代码，客户可以自己添加直接的协议）；NL6621 NuAgent SDK 在系统上电启动时，先判断 DirectConfig 是否启用。

4.3.1 SoftAP

SoftAP 配置信息

热点名称（SSID）：NuAgent-XXXX，XXXX 是 mac 地址后四位

密码：123456789

IP 地址：10.10.10.1

UDP 配置端口：60001

SoftAP 配置过程：

- 1) 设备第一次启动，默认进入 softap 模式；
- 2) 等待 APP 端的连接，接收 SSID 以及 Password，并且换到 STA 模式并连上 AP；
- 3) 设备连接成功，则保存 SSID 以及 Password 到 norflash 并重启系统。
- 4) 系统重新启动之后，从 norflash 中读取保存的 SSID 以及 Password 并连接到 AP；

注：如果之前保存的 AP 无法连接，系统会尝试连接 5 次，5 次连接失败后，切换回 SoftAP 模式。

4.3.2 DirectConfig

DirectConfig 配置信息

接收方式：组播

DirectConfig 配置过程

- 1) 系统上电启动时，按下 DirectConfig 按键 2s
- 2) LED 指示灯以 200ms 间隔进行闪烁为进入 DirectConfig 模式。
- 3) 获取到 SSID 以及 Password 之后开始连接 AP，以 1s 间隔进行闪烁。
- 4) 连接成功后 LED 指示灯灭。并保存获取到的 SSID 以及 Password。

注：与 SoftAP 配置模式一样，系统尝试连接 5 次，5 次连接失败后，切换回 SoftAP 模式。

4.4 Uart 收发数据

任务线程 UartTaskThread 用于完成接受从串口发来的数据，代码中已经屏蔽相关线程（由于 NL6621 只有一个串口，因此如果串口用于特殊功能，那么用户将不能使用正常的 Debug 打印功能）。

4.4.1 串口收发机制

NL6621 发送数据通过 `int uart_data_send(unsigned char *data, unsigned short len)` 接口发送数据。参数 `data` 为发送数据的 buffer，`len` 为发送数据的长度。

NL6621 接收串口数据的 buffer 大小设为 512，当接收到的串口数据超过 256 或者接收数据的时间超过 100ms 则唤醒串口接收 UartTaskThread 任务线程。在 UartTaskThread 线程中完成用户串口数据的处理。用户需要在该任务线程中处理相关数据。

4.5 模拟串口功能说明

NuAgent 提供模拟串口，经过测试在波特率 9600,14400,19200,38400 较稳定，默认使用波特率 38400，发送数据默认使用 GPIO5，接收数据默认使用 GPIO6；log 日志如果需要使用模拟串口输出，需要在 log.h 文件 `REAL_UART_USED` 宏定义屏蔽。模拟串口主要涉及到的源文件 `simu_uart.c`、`simu_uart.h`、`log.c`、`log.h`。

模拟串口发送设计思路：采用 FIFO 来保存将要发送数据，在定时器 1 中断轮询 FIFO 是否有数据，如有数据定时器采用 10 次中断发送一个数据，如此反复。

模拟串口接收设计思路：使用定时器 0 和 GPIO 中断结合使用，设置 GPIO 中断只在判断串口开始位是使能，设置下降沿中断，在 GPIO 中断函数设置定时器 0 溢出时间是波特率一半，确保有效采集 8 个电平，如此反复。

4.6 Sniffer 一键配置流程

Sniffer 配置模式思路：首先调用 `snifferconfig_init` 配置初始化配置参数，获取附近路由器，统计路由器所在热点的信道，得到需要扫描 AP 的信道列表，每隔 0.3 秒切换信道，获取到附近的广播包后，调用 `skyconfig_process`，如果 `snifferconfig_process` 返回的 `snifferconfig_status` 是 `Running_Lock`，则立刻锁定当前信道不再每隔 0.3 秒切换；如果调用 `snifferconfig_process` 返回的 `snifferconfig_status` 是 `Finish_OK`，则表明已经接收完成。

目前该配置代码，只提供代码流程，协议需要自己添加，需要添加的接口如下：`snifferconfig_init()`、`snifferconfig_process()`、`get_sniffer_wifi_info()`。

4.7 BSP 外设接口 Demo 测试

NuAgent 提供的 BSP 源代码一大部分采用 STM32 库编程方式，每个外设都有相应的 Demo（NL6621_BSP_TEST），在 `BspDemo.h` 使用宏定义使能各个外设测试，避免用户使用繁琐的寄存器编程，方便用户快速开发使用。

4.7.1 GPIO 设置输出输入模式

NL6621 的 GPIO 资源：

GPIO	管脚
GPIO3	PIN48
GPIO4	PIN50
GPIO5	PIN51
GPIO6	PIN52
GPIO7	PIN53
GPIO8	PIN47
GPIO9	PIN07
GPIO10	PIN08
GPIO11	PIN16
GPIO12	PIN17
GPIO17	PIN19
GPIO18	PIN21
GPIO19	PIN22
GPIO20	PIN20
GPIO21	PIN09
GPIO22	PIN10
GPIO23	PIN11
GPIO24	PIN13
GPIO25	PIN14
GPIO26	PIN15
GPIO27	PIN03
GPIO28	PIN04
GPIO29	PIN05
GPIO30	PIN06
GPIO31	PIN23

使能 GPIO_Demo 宏定义，默认使用 GPIO5 输出高低电平，GPIO3 设置输入模式，在测试 GPIO_Demo 时，统筹系统所使用到的 GPIO，防止冲突使用。

4.7.2 GPIO 中断模式

使能 GPIO_EXTI_Demo 宏定义，NL6621 支持 25 个 GPIO 口设置成中断 GPIO 口（除了 GPIO0\GPIO1\GPIO2\GPIO13\GPIO14\GPIO15\GPIO16）。用户配置 GPIO 管脚为外部中断，设置电平或者上下沿触发，配置方式参考 GPIO_Interrupt_Demo、BSP_GPIOIntISR_Demo 接口。

4.7.3 定时器模式

使能 `TIMER_Demo` 宏定义，NL6621 支持 3 个定时器，其中一个滴答时钟已被 UCOS 使用，还剩下 2 个硬件定时器，使用方法可以参考 `Timerx_Init()`、`TMR1_IRQFunc()`、`TMR0_IRQFunc()` 接口；

1. 定时器初始化，如果使用模拟串口的话，定时器 0,1 都被使用了，定时器 1 用于串口发送，定时器 0 用于串口接收。
2. 如果用到 SPI 传输数据，使能 `SPI_SDIO_CMD_TEST` 的话，定时器 1 已经使用。
3. 定时器中断范围（1us~107374182us(107s)）。

4.7.4 看门狗

使能 `WATCH_Demo` 宏定义，看门狗溢出时间只能取 100ms,200ms,400ms，`WATCH_Feed_Demo` 喂狗，必须提前 20ms 喂狗。具体使用模式可以参考 `IWDG_Init_Demo()`、`WATCH_Feed_Demo()` 接口说明。

4.7.5 I2C 读写 EEPROM

使能 `I2C_EEPROM_Demo` 宏定义，反复读写 EEPROM；具体参考 `eeeprom_init()`、`eeeprom_data_write()`、`eeeprom_data_read()` 接口。

4.7.6 QSPI 读写 FLASH

使能 `QSPI_FLASH_Demo` 宏定义，已经在 BSP 初始化验证过。

4.7.7 SPI 传输

使能 `QSPI_FLASH_Demo` 宏定义，SPI 初始化，采用 SDIO CMD53，`SPI_CS` 使用模拟 GPIO，详细参考《NL6621 SDK 用户手册》第 3.10 单节。

4.7.8 I2S 读写数据

使能 `HW_I2S_SUPPORT` 宏定义，提供 I2S 发送和接收，采样位数 24bit 采样频率 48KHZ、飞利浦格式。

4.7.9 SDIO 读写数据

使能 QSDIO_TEST 宏定义，SDIO 不能作为 Host，只能做 device，采用 SDIO CMD53 与 SPI 互相传输数据，详细参考《NL6621 SDK 用户手册》第 3.10 单节。

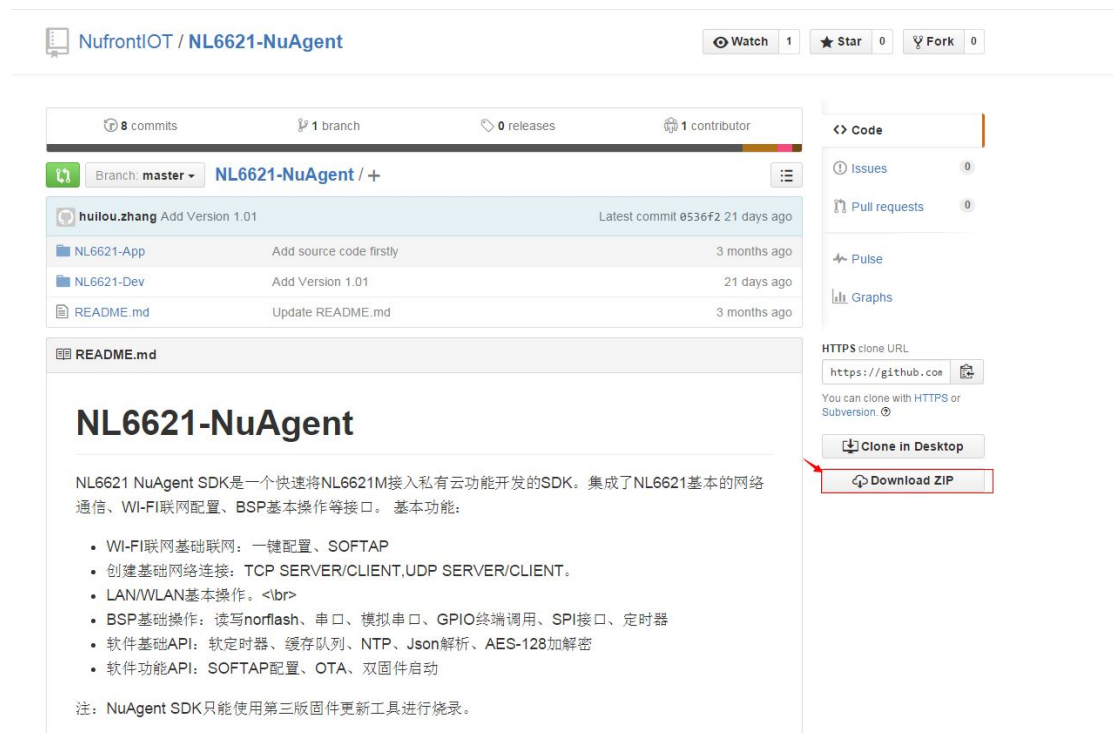
4.7.10 DMA 内存搬运

使能 DMA_MOVE_MEM 宏定义，采用 DMA 传输，DMA 传输只能是 32bit。接口：NST_DmaMoveMem。

5. NuAgent SDK 获取

NuAgent SDK 的相关代码已经更新到 github，后续对 NuAgent 的代码更新以及维护都将在 github 上进行。相关代码连接资源如下所示：

<https://github.com/NufrontIoT/NL6621-NuAgent>



NufrontIoT / NL6621-NuAgent

Watch 1 Star 0 Fork 0

8 commits 1 branch 0 releases 1 contributor

Branch: master NL6621-NuAgent +

huiyou.zhang Add Version 1.01 Latest commit 0536f2 21 days ago

File	Commit Message	Time
NL6621-App	Add source code firstly	3 months ago
NL6621-Dev	Add Version 1.01	21 days ago
README.md	Update README.md	3 months ago

README.md

NL6621-NuAgent

NL6621 NuAgent SDK是一个快速将NL6621M接入私有云功能开发的SDK。集成了NL6621基本的网络通信、Wi-Fi联网配置、BSP基本操作等接口。基本功能：

- Wi-Fi联网基础联网：一键配置、SOFTAP
- 创建基础网络连接：TCP SERVER/CLIENT,UDP SERVER/CLIENT。
- LAN/WLAN基本操作。

- BSP基础操作：读写norflash、串口、模拟串口、GPIO终端调用、SPI接口、定时器
- 软件基础API：软定时器、缓存队列、NTP、Json解析、AES-128加解密
- 软件功能API：SOFTAP配置、OTA、双固件启动

注：NuAgent SDK只能使用第三版固件更新工具进行烧录。

Code

Issues 0

Pull requests 0

Pulse

Graphs

HTTPS clone URL

<https://github.com>

You can clone with HTTPS or Subversion

Clone in Desktop

Download ZIP

6. 注意事项

6.1 库的使用

在 NuAgent 的 SDK 中，由于 NL6621 使用的代码空间有限，对 ANSI C 中的字符处理函数等做了相应的裁剪，用户如需要使用到这些函数，可以再 `util.h` 文件中寻找，或者在 `str_lib.c` 文件中寻找，如果找不到，那么需要用户自己移植相应的字符串接口。