



深圳市海凌科电子有限公司

HLK-N10 平台开发指南

使用前说明

本书采用的约定及标志如下。

1. 符号约定

带尖括号“< >”表示键名、按钮名以及操作员从终端输入的信息；带方括号“[]”表示人机界面、菜单条、数据表和字段名等，多级菜单用“→”隔开。如 [文件→新建→文件夹] 多级菜单表示 [文件] 菜单下的 [新建] 子菜单下的 [文件夹] 菜单项。

2. 键盘操作约定

格式	意义
加尖括号的字符	表示键名、按钮名。如 < Enter >、< Tab >、< Backspace >、< a > 等分别表示回车、制表、退格、小写字母 a
< 键 1+键 2 >	表示在键盘上同时按下几个键。如 < Ctrl+Alt+A > 表示同时按下“Ctrl”、“Alt”、“A”这三个键
< 键 1, 键 2 >	表示先按第一键，释放，再按第二键。如 < Alt, F > 表示先按 < Alt > 键，释放后，紧接着再按 < F > 键

3. 鼠标操作约定

格式	意义
单击	快速按下并释放鼠标的左键
双击	连续两次快速按下并释放鼠标的左键
右击	快速按下并释放鼠标的右键
拖动	按住鼠标的左键不放，移动鼠标

4. 标志

小心、注意、警告、危险前使用符号“”。

说明、提示、小窍门前使用符号“”。

目录

1 概要	1
2 平台说明	1
2.1 SDK 特征介绍	1
2.2 系统资源	1
2.3 外设资源	2
2.4 主控平台框架介绍	2
2.5 典型流程介绍	3
2.6 文件目录介绍	4
2.7 平台功能集说明	4
3 用户开发指导	5
3.1 用户二次开发总则	5
3.2 用户二次开发注意事项	5
3.2.1 出厂 NV 重要参数	5
3.2.2 close_debug 参数使用说明	7
3.2.3 关于 malloc/free 的使用说明	7
3.2.4 代码段存放 RAM 和 flash 的说明	8
3.2.5 平台 lib 库使用原则	8
3.3 AT 开发指导	8
3.3.1 特征介绍	8
3.3.2 AT+WORKLOCK 工作锁使用说明	8
3.3.3 8007 AT 通道报错	9
3.3.4 低功耗模式 AT 命令开发建议	9
3.3.5 注意事项	10
3.4 省电开发指导	10
3.5 M3 单核启动开发指导	10
3.5.1 不唤醒 DSP 核的单核工作流程	10
3.5.2 动态唤醒 DSP 的工作流程	10
3.5.3 注意事项	11
3.6 NV 与 FLASH 的使用	11
3.6.1 用户 flash 使用说明	11
3.6.2 用户出厂 NV 的使用建议	12
3.6.3 NV 使用说明及异常场景处理介绍	13
3.6.4 OPENCPU 秒级别数据保存方案设计	13
3.6.5 NV 及 flash 的调试	14
4 DEMO 实例使用说明	15
4.1 用户任务开发指导	15
4.2 DEMO 功能列表	15
4.3 DEMO 开发使用说明	18
4.4 用户二次开发文件介绍	18
版本历史	20

1 概要

本文描述的是海凌科 NBIOT 芯片的软件平台开发指导，供读者进行用户程序的开发，包括网络应用、扩展 AT、设备操作等。文档适用于用户系统工程师，软件开发工程师和产品工程师等相关技术人员查阅使用。

2 平台说明

2.1 SDK 特征介绍

海凌科 SDK 主要特性：

- eclipse 集成开发调试环境，配套有支持文档和调试工具
- 华为 LiteOS(V200R001C50B021)，详细信息查看《Huawei_LiteOS_Developer_Guide.pdf》
- 支持实现 AT /Socket/多种云 API，调用简单灵活
- 提供各种网络业务 lib 库，支持用户通过编译宏方式选择需要的 lib 库
- 扩展 AT 命令注册机制，方便用户定制扩展私属 AT
- 灵活的 flash 操作接口，支持磨损均衡；
- 支持外部 MCU 对芯片断电，确保上电唤醒后流程最优，功耗更低；
- 灵活的 PSM 省电策略，功耗更低；
- 快速 RAI 流程技术突破，功耗更优；
- 支持硬件加密机制，保证通信的安全可靠性
- 紧凑型高性能的预编译代码库
- 灵活可编程的 PIN 脚资源
- 丰富的示例工程

2.2 系统资源

主控核为 32-bit ARM Cortex-M3 RISC，主频最高 106M，用户理论可用最高 130MIPS。

主控核使用的是 Lite-OS 操作系统，版本号为 LiteOSV200R001C50B021。

芯片内部集成 900K 字节 RAM，主控核 M3 用户可用的 RAM 空间约为 100K，没有外部 RAM。

芯片使用型号为 GD25Q16 的 flash，2M 字节空间。目前用户直接可用空间约为 160K 左右，若超过，则需要抢占 FOTA 差分包的 flash 空间，FOTA 升级期间会被擦除，约 210K 字节。

平台支持 FOTA 差分升级，差分包上限为 210K 字节。

2.3 外设资源

芯片从 standby 状态唤醒，需等待约 3 毫秒后软件系统方可用；从 DEEPSLEEP 深睡唤醒，需等待约 35 毫秒软件系统方可用。

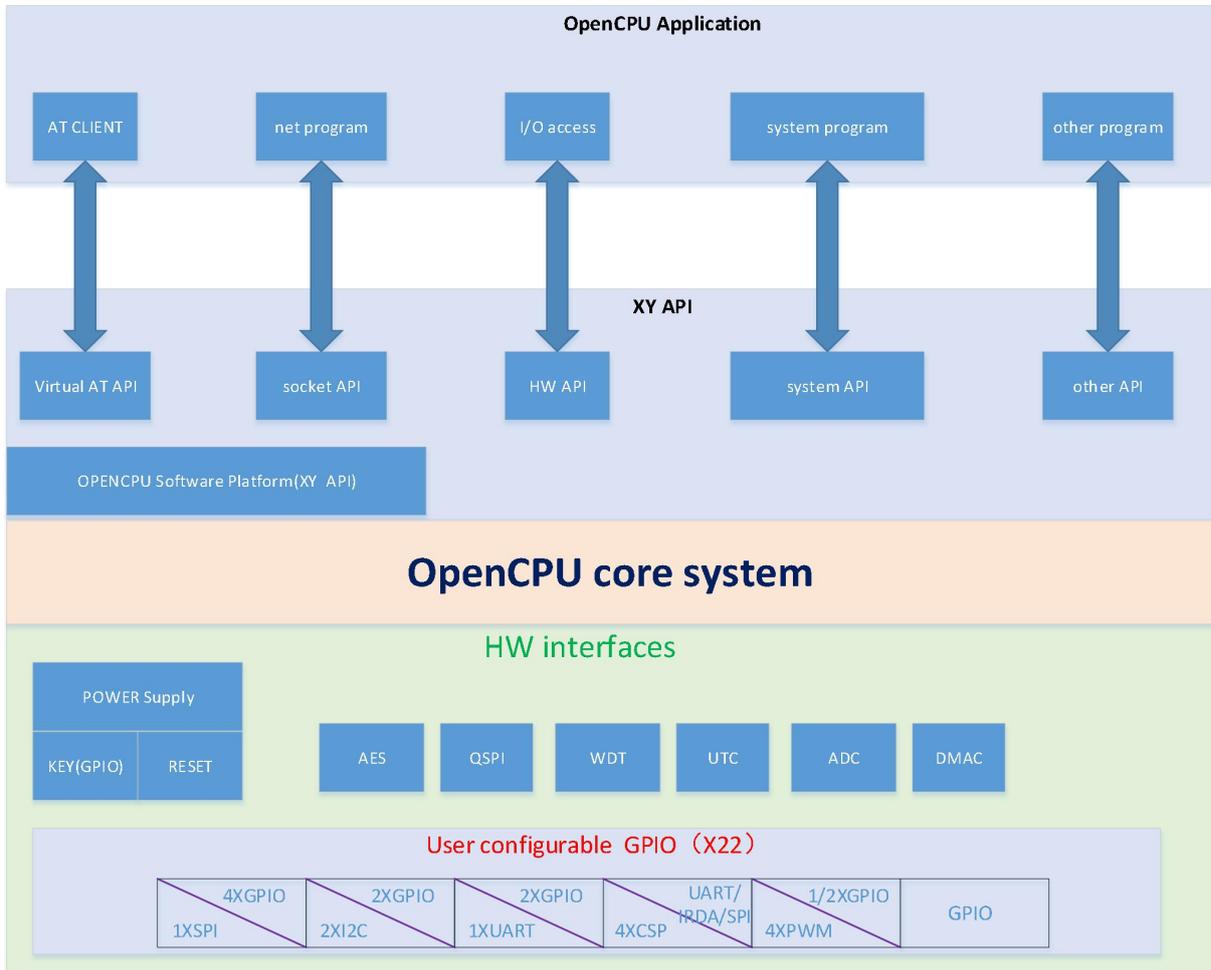
UART 支持 2400 到 921600 波特率配置，有两个 UART 可用于二次开发，暂无 LPUART。由于 STANDBY 唤醒时稳定 PLL 需要 3 毫秒左右，存在数据出错风险。

GPIO-10/11/12 仅支持内部下拉，其他的 GPIO 仅支持内部上拉。

ADC 位数为 12 位，单次采样时间 5us。

外设资源详细信息，请参阅《HLK-N10_外设使用说明》。

2.4 主控平台框架介绍



OpenCPU 平台是软硬件一体的模块思想，包含 ARM-M3 微处理器，使用 LiteOS 实时操作系统，提供实时多任务调度能力。

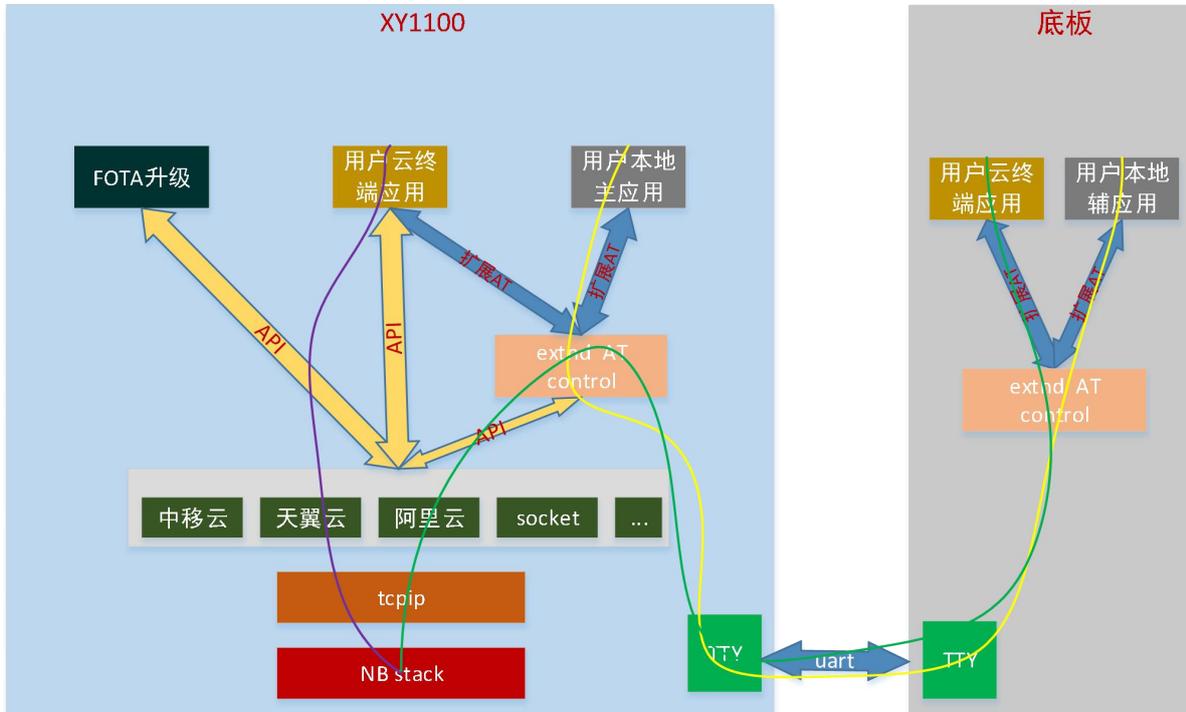
OpenCPU 平台提供 AT 接口，供用户控制 NB 协议栈；同时，平台提供开放的注册接口，供用户自定义扩展 AT 命令。

OpenCPU 平台提供动态启动云网络业务线程的能力，当用户需要使用时，调用 API 动态创建启用。

OpenCPU 平台提供 I/O 接口，供用户控制硬件资源及外设。

OpenCPU 平台进一步开放可灵活配置的 GPIO 接口，供用户进行 UART/SPI 等外部总线的组合设计。

2.5 典型流程介绍



上图的框图描述的是软件架构图，软件平台提供了 OS/AT/多种云/SOCET/DEVICE 等封装接口；同时，为了用户底板开发的方便，海凌科提供了自定义的 AT 命令集。

软件平台提供了两种软件参考，一种是带用户底板的产品形态，用户主要在自己底板上进行业务开发，并通过扩展 AT 命令与 HLK-N10 进行通讯。上图中黄色流线描述的是用户底板上的用户本地应用通过扩展 AT，经 UART 等通道，与 HLK-N10 上的用户本地应用进行交互的流线；上图中绿色流线描述的是底板的用户云终端应用通过扩展 AT 命令，与 HLK-N10 上的扩展 AT 云代理模块进行信息交互，进而通过 NB 协议栈与远端服务器进行网络通信。

另一种是不带底板的产品形态，用户所有的软件功能位于 HLK-N10 上。上图中紫色流线描述的是用户云终端应用调用具体云终端的 API 接口，经 TCPIP 和 NB 协议栈，与远端服务器进行网络通信。

2.6 文件目录介绍

SDK 提供一整套 API 接口,供用户开发使用,用户仅需关注 userapp 下的代码即可,不得使用非 userapp 下的任何接口、宏定义、全局等。

文件夹	功能介绍	使用说明
arch	cortex-M3 系统的编译配置文件	不建议用户修改
examples	liteos 操作系统各个接口的实例	不参与编译, 仅供代码参考
kernel	liteos 操作系统核心代码	不建议用户修改
sys_app	平台业务基础功能代码, 包括 AT、FTL、time、RTC、核间通信等	不建议用户修改
target	多种 drive、demo 等, 同时也包含 makefile 功能文件夹	不建议用户修改
TCPIP	lwip 协议栈、ctwing、onenet..	用户通过编译宏开关所有功能, 但不能在内部修改代码, 必须放在 userapp 中进行自己云平台的开发
userapp	用户 demo 参考代码及用户可用的头文件声明	用户仅需 include "softap_api.h" 即可

2.7 平台功能集说明

请参阅《海凌科 SDK_release_note.xlsx》功能清单表项。

3 用户开发指导

3.1 用户二次开发总则

用户开发的代码，必须位于 userapp 文件夹下，其他路径不能存放。

userapp 文件夹下的 inc 头文件，供用户调用的接口声明，用户不得进行修改，若需要，请单独新增用户头文件。

用户若使用 M3 核的平台业务功能，如 socket、OneNET、CDP 等，只能通过 API 接口，不得使用扩展 AT 命令接口。

用户代码除非有特别需求，否则一律放入 flash 中运行，不得放入 RAM 中。

对于打印、内存管理等支撑相关类接口，用户不得直接使用操作系统接口，统一使用 os_adapt.h 中定义接口。

平台提供了硬看门狗机制，确保软件正常调度运行。对于用户任务的异常，用户需要自行设计软看门狗机制，如规定时间内未能与网络完成通信等，则触发重启等策略。

3.2 用户二次开发注意事项

3.2.1 出厂 NV 重要参数

参数名	变量名	Debug 值	默认值	含义	使用说明
VERTYPE	ver_type	2	2	位图表示业务平台的版本功能。 1 为入库版本； 2 为通用 SDK 版本； 4 为金卡专用版本； 8 为中移版本；	对于部分用户提出的与平台默认功能相左的实现方案，平台通过该位图来区分不同的实现，如果用户没有提出特别需求，默认值即可。
WORKMODE	work_mode	0	0	设置上电启动时的工作模式。 0 表示双核启动，常见于外部 MCU 产品形态； 1 表示仅启动 M3 核，后续根据需要再动态启动 DSP 核，常见于 OPENCPU 产品形态。	对于 OPENCPU 产品形态，如烟感报警器，正常工作时仅 M3 上电，当需要进行网络通信时，再动态的加载 DSP 核。
DEEPSLEEP	deepsleep_enable	1	1	设置深睡模式的开关。 0 表示关闭，常见于长供电场景，WORKLOCK 无效； 1 表示打开，用于功耗敏感型产品。	若用户产品为长供电产品，则设置为 0，不支持深睡，并且 WORKLOCK 无效。
IPALIVE	ip_keep_alive	0	0	设置是否进行 IP 地址保活，即是否支持 TAU 或 eDRX。 0 表示无需 TAU 周期性更新来保活 IP 地址，目前市面上的产品皆属于此形	芯片将按照 3GPP 的 PSM 协议进入深睡，无需输入 AT+WORKLOCK=0；工作期间若用户需要执行事务，需要输入 AT+WORKLOCK=1 锁住，否则

				态; 1 表示需要进行远程下行控制, 需保活 IP 地址及链接, 目前运营商尚不支持长时间保活链接。	会自动进入深睡, 造成用户业务被打断
DOWNDATA	down_data	1	1	1 表示释放锁后存在下行突发数据可能, 如远程开关阀门, 远程查询等事务, 则 AT+WORKLOCK=0 后, 可以继续处理下行数据包; 0 表示释放锁后不存在下行突发数据, 输入 AT+WORKLOCK=0 后, 不再处理下行数据包;	对于终端触发的数据通信产品, 如果确保收完下行数据后才会执行深睡操作, 可将该值设为 0, 以加快深睡流程, 降低功耗。
OFFTIME	offtime	0	0	开关 OFFTIME 时间补偿, 0 表示关闭, 断电上电后协议栈执行 attach; 1 表示打开, 断电上电后需要通过 AT+OFFTIME 命令进行断电时长补偿, 供平台决定协议栈动作	OPENCPU 产品只能为 0
BACKUPKEEP	keep_retension_mem	0	0	是否保持 retension memory 不断电。该 4K 内存保存了平台的易变 NV, 每次工作都会发生变化。对于不支持断电模式的产品, 可以设置该值为 1, 以减少写 flash 的次数, 避免坏块风险	如果产品使用周期小于 1 小时, 建议设为 1, 以防止 flash 坏块的风险。设为 1, 深睡模式下, 该内存会额外增加 0.3uA 的电流开销。
CLOSEURC	close_urc	0	1	1 表示过滤掉底板 MCU 不需要的一些主动上报 AT 命令 URC, 以降低功耗开销	用户可以在 drop_unused_urc 接口中添加自己不感兴趣的 URC 过滤, 以降低功耗
UARTSET	uart_rate	4	4	芯片默认 AT 串口波特率为 9600, 用户可以修改, 以 2400 倍数方式赋值	支持通过“AT+UARTSET”动态修改波特率, 但需要通过“AT”“OK”进行握手
LOG	open_log	1	0	芯片 log 的开关, 0 表示关闭所有 log; 1 表示打开所有 log; 2 表示仅输出 M3 核 log; 3 表示仅输出 M3 核上 xy_printf 接口对应的 log, 即用户 log。	为了降低功耗, 减少不必要内存开销, 建议用户关闭 log 输出; 若调试需要, 可通过 AT 命令动态打开
CLOSEDEBUG	off_debug	0	1	debug 调试开关, 产品化测试时关闭	生产时必须关闭该值
SIMVCC	sim_vcc_ctrl	0	0	SIM 卡供电参数; 0 表示仅支持 3 伏; 1 表示仅支持 1.8 伏; 2 表示自适应, 优先 1.8 伏	若用户为 XO 晶振, 且需要 SIM 卡自适应功能, 设为 2 即可。
	resetctl	1	0	芯片上的复位 / 唤醒引脚, 0 表示高电平大于 20 毫秒为复位, 小于 20 毫秒为唤醒; 1 表示高电平大于 6 秒为复位, 小于	用户根据产品形态和研发阶段, 通过 AT 命令: AT+RESETCTL=<val>来进行配置。

				6秒为唤醒。	
BACKUPT HRESHOLD	backup_threshold	0	0	当 keep_retension_mem 为 1 时，配置该阈值来实现根据深睡时长动态控制 retension 内存断电与否。当深睡时长小于该阈值时，保持供电；当大于该阈值时，回写 flash 并断电。	该阈值仅用于 eDRX 产品形态，不得在其他产品，一般建议设置为 30 分钟，即 1800 秒
OPENTAU URC	open_tau_urc	0	0	由用户决定 TAU 唤醒后是否上报命令给底板 MCU。0 表示不上报	由于 TAU 周期性行为属于芯片内部行为，不建议上报给底板 MCU
RAI	close_rai	0	0	指示是否关闭平台的释放锁后自动触发 RAI 的能力，1 表示关闭。	设 1 后，用户必须自行触发 RAI 流程，加快链接的快速释放，以降功耗
RTCURC	open_rtc_urc	0	0	指示用户 RTC 唤醒后，是否触发主动上报通知底板 MCU。 0 表示不上报 URC	1 表示底板 MCU 将会唤醒工作。
MAH	min_mah	0		FOTA 升级所需的最低电池容量，若小于该容量，则放弃升级	电池电量需要由用户提供函数具体实现
VBAT	min_mVbat	0		芯片正常工作所需的最低电池电压，若小于该值，则不能进行 flash 的擦写操作，以防止数据异常	接口实现考虑了抖动情况

备注:

- 用户通过 AT+NVS=SET,<param>,<val>进行参数值设置，且必须 AT+NRB 重启后 NV 设置值才会生效。通过 AT+NVS=GET,<param>获取 NV 参数值，以检查是否配置正确。AT+NVS 的命令仅用于调试阶段。
- 为了便于调试开发，对某些 NV 及流程添加了 debug 功能的圈定设计，在生产前必须将 close_debug 值改为 1，并进行充分的遍历验证。
- 如果用户产品不采用断电模式进行省电，必须将 offtime 设置为 0，否则异常断电后，必须输入命令“AT+OFFTIME=0”才能正常 Attach。
- 由于运营商目前不能很好的支持 eDRX，若用户产品需要修改 ip_keep_alive、down_data、offtime 三个默认值，必须与海凌科 AE 联系。
- 关于低功耗相关的配置，请阅读《海凌科 HLK-N10 低功耗开发指南》。

3.2.2 close_debug 参数使用说明

debug 功能用于研发调试阶段，对一些异常进行定位跟踪，例如软件运行异常的断言保持现场等。在生产前必须将 off_debug 值改为 1，并进行充分的遍历验证。用户通过 AT+NVS=SET,CLOSEDEBUG,1 命令来关闭 DEBUG 功能，以进行产品化测试。

AT+NVS 命令是为了方便用户调试而设计的，生产后该命令不再使用，所以 NV 参数的修改必须在生产时通过 NV 工具进行烧录，不得把 AT+NVS 作为正常的 AT 代码进行开发使用。

3.2.3 关于 malloc/free 的使用说明

由于 lib 库中提供的 malloc 和 free 存在自己的内存管理机制，与 liteos 的内存管理存在冲突，所以用户二次开发时，皆必须使用 xy_malloc 和 xy_free 进行内存申请释放，不能直接使用 malloc 和 free。

3.2.4 代码段存放 RAM 和 flash 的说明

芯片仅含一个 flash 控制器，且两个 CPU 皆会执行 XIP 模式，为了确保系统的正常运行，需要遵循以下原则：

- 支撑类及高概率调用的函数，因涉及到性能，建议放 RAM 中，用户代码放 flash 中运行；
- 由于 DSP 核工作期间需要 XIP 运行代码，所以不支持 M3 核工作态过程中进行 flash 的写操作，除非确定 DSP 核进入睡眠模式；
- 放 Flash 上运行的代码无法打断点在线调试，若需要断点调试，请临时移到 RAM 空间运行；

3.2.5 平台 lib 库使用原则

平台目前提供了 Socket、OneNET、CDP 等网络业务 lib 库，用户可通过修改 `feature.mk` 配置文件来选择需要的业务功能模块，具体参看《海凌科 HLK-N10 Gcc 编译与 eclipse 调试指导说明》。

用户开发的代码，必须位于 `userapp` 文件夹下，其他路径不能存放。

由于 AT 代理平台与省电、串口驱动紧藕，不建议用户替换自己的 AT 业务平台。

由于 FOTA 升级与平台的底层软件、分区等紧藕，不建议用户替换自己的 FOTA 升级。

由于 LWIP 协议栈底层与驱动紧藕，不建议用户替换自己的 TCPIP 协议栈。

建议用户可以实现 Socket、CTWing、OneNET、CoAP、LwM2M 等自有业务功能，相关的 AT 命令需要注册到平台的接口中。

3.3 AT 开发指导

3.3.1 特征介绍

海凌科提供的 AT 指令固件具有以下特色，利于用户二次开发：

- 外在的 API 接口，供用户进行业务功能的极简开发
- 平台内部始终维护 NB 网络可用，减少用户的 AT 开发工作
- 内置 TCP/IP 堆栈和数据缓存
- 内置天翼云、OneNET、电信云堆栈和数据缓存
- 能便捷地集成到资源受限的主机平台中
- 主机对指令的响应易于解析
- 用户可自定义 AT 指令

平台提供了 API 接口供用户进行二次开发，包括标准 3GPP AT 命令的使用及扩展 AT 命令的使用，具体使用请参看 demo 中的实用例。平台提供的扩展 AT 命令集，作为芯片的内在功能，可以在平台的 ARM 核上使用，也可以由用户底板 MCU 使用。

3.3.2 AT+WORKLOCK 工作锁使用说明

WORKLOCK 工作锁的目的是在用户使用芯片期间，禁止芯片进入深睡，以确保用户流程不被打断，从而提高网络数据传输的实时性。

为了支持多用户独立使用芯片通讯能力，要求每个模块需要使用芯片时申请锁，不再使用时释放锁，即必须配对使用锁机制。

具体使用指导，请阅读《海凌科 HLK-N10 低功耗开发指南》。

3.3.3 8007 AT 通道报错

8007 是通道 BUSY 报错。原则上，一条 AT 请求未被应答之前，再发送下一条 AT 命令，会报错。因为一旦后续回复应答结果，无法识别是上一条命令还是当前命令的请求。

用户调试过程中，若遇到一直回复 8007 的，则表明**第一条未收到答复的 AT 命令**流程异常，无法回复结果码，AT 口会收到“+DBGINFO:BUSY:”的主动上报 debug 信息，以告知是哪条命令正在处理。

另外，由于 DSP 单核死机，ARM 核依然能正常工作，此时如果发送 AT 命令给 DSP 核，则无法收到应答，进而出现 8007 报错。

目前，除了下表中 AT 命令工作时长达分钟的，其他的 AT 命令皆应该能够 5 秒内收到应答结果。若不能够按照期望收到应答结果，则需要定位问题。产品化阶段，对于超时仍然无法收到正确应答结果的，建议重启设备，无需再重发 AT 命令了，重发仍然会上报 8007 错误。

耗时	命令
网络决定	+XSOPEN、+XSCLOSE、+XDNS
~10s	+CPIN、+CLCK、+CPWD、AT+CFUN=0
>60s	AT+CGATT=1、AT+CGACT=1、AT+CGCMOD=0、AT+CMGS
~5min	AT+COPS=1/0

3.3.4 低功耗模式 AT 命令开发建议

由于 STANDBY 睡眠机制会关闭 AT 串口的 BBPLL 时钟，当下次中断唤醒后 BBPLL 稳定需要约 3 毫秒左右。深睡唤醒后系统需要重新上电及初始化，待 AT 通道可用，需要大约 30 毫秒时长。

针对低功耗模式的使用，推荐如下方案：

- 深睡模式下，外部 MCU 通过触发 WAKEUP_PIN 可唤醒中断，唤醒芯片，待收到“+POWERON:”主动上报后，再发送 AT 命令给芯片；
- 版本默认串口波特率为 9600bps，在此波特率下，通过软件机制解决 STANDBY 模式唤醒时 BBPLL 稳定之前的脏数据问题，确保 AT 命令正常收发；
- 由于软件机制无法解决高波特率(大于 19200bps)下 STANDBY 唤醒时丢 AT 命令情况，建议用户默认使用 9600bps 波特率，针对大数据传输需要更高波特率，建议使用“AT+UARTSET”扩展 AT 命令进行动态波特率更改，且内部自动关闭 STANDBY 睡眠，以减少数据传输时长；
- 由于动态波特率切换过程中，外部 MCU 与芯片存在时间差，建议切换完后，通过“AT”“OK”方式进行握手确认。

3.3.5 注意事项

因平台未对栈空间中的局部变量进行初始化，所以每条 AT 命令的解析参数，需要进行初始化赋值，否则会造成流程紊乱。

波特率默认 9600bps，由于 standby 睡眠唤醒采用了软采样技术，要求发送的 AT 命令必须大于等于 4 字节，且第三个字节必须是 g_AT_Req_third_chars 全局数组中其中一个字符，用户可以根据自己需求定制不同的 AT 前缀。

由于低功耗对串口产生影响，进而会造成 8003 错误，请仔细阅读《海凌科 HLK-N10 低功耗模式下数据传输指导说明》。

3.4 省电开发指导

参见文档《HLK-N10_低功耗开发指南》。

3.5 M3 单核启动开发指导

平台支持仅 M3 核上电场景，以满足用户周期性检测的场景。建模场景为：不带底板 MCU 场景，传感器挂接在 M3 主控核上，M3 核以若干分钟为周期唤醒工作，读取传感器数值后，交由 M3 核用户线程决策是否触发远程网络数据通信；若决策需要远程网络通信，动态唤醒 DSP 核，进行数据通信即可。

目前平台在实现上，充分考虑了双核和单核两种工作场景的兼容，用户二次开发仅存在少许的差异，即若需要动态唤醒 DSP，需要发送空 AT 唤醒 DSP 核。

3.5.1 不唤醒 DSP 核的单核工作流程

```
+POWERON:0 //断电上电、UTC唤醒、PIN中断唤醒等主动上报
```

```
AT+WORKLOCK=1
```

```
OK
```

```
//用户按照触发事件执行相关本地动作
```

```
AT+WORKLOCK=0
```

```
OK
```

```
+POWERDOWN:-1 //芯片进入深睡
```

3.5.2 动态唤醒 DSP 的工作流程

```
+POWERON:0 //上电开机、UTC唤醒、PIN中断唤醒等主动上报
```

//用户按照触发事件执行相关本地动作，决策需要远程网络数据通信

AT+WORKLOCK=1 //申请双核锁

OK

AT //该空命令用于动态启动 DSP 核

OK

//按照双核流程进行远程数据通信

AT+WORKLOCK=0 //完毕后，释放锁

OK

+POWERDOWN:-1 //芯片进入深睡

3.5.3 注意事项

为了简化用户操作，单核模式不支持 OFFTIME 时间补偿，每次断电后重新上电开机，若动态启动 DSP 核，皆会进行 attach。仅允许空 AT 命令("AT\r\n")动态唤醒 DSP 核，其他命令一律上报 "+CME ERROR:8002"，即操作不允许。

OPENCPU 产品形态，用户在 M3 核上进行单核开发，仅需调用 xy_work_lock()即可，平台内部会执行动态启动 DSP 核流程。

3.6 NV 与 flash 的使用

3.6.1 用户 flash 使用说明

目前，平台使用的 Flash 是兆易公司生产的 NOR-flash，擦写上限次数为 10 万次，以 10 年使用寿命来计算，擦写频率**不能小于 1 小时一次**。Flash 的 sector 扇区单位为 4K 字节，擦除或写入一次耗时 **50 毫秒**左右；读取很快，小于 1 毫秒。

平台提供了从 USER_FLASH_BASE 开始，大小为 USER_FLASH_LEN_MAX 字节的 flash 空间供用户保存数据到 Flash 中。芯片支持工作期间实时写 Flash，但是由于执行写 Flash 操作时，会退出 XIP 模式，进而不能运行 Flash 上的代码。所以要求所有的中断服务程序中不得调用 flash 上的代码。可以用 `__attribute__((section(".ramtext")))`方式把某些函数放在 RAM 上。

由于 Flash 擦除过程中异常断电，会造成整个扇区值的异常，进而需要进行异常断电保护容错。平台针对 NV 数据进行了异常断电的保护容错，确保系统的正常运行。针对用户使用的用户 flash 空间，即 USER_FLASH_BASE 开始的区域，需要用户考虑异常断电的容错。建议每次写 Flash 时，在每个扇区的头部添加 4 个字节的魔术数字，如 5A5A5A5A；对每个扇区的尾部添加 4 个字节的 checksum 校验值，可以调用 xy_chksum 接口来获取校验值。每次读取 Flash 时，需要检查魔术数字和 checksum 是否正确，若不正确，则认为 flash 内容异常，不能使用。

平台提供了回调机制，供用户自行决定 USER_FLASH_BASE 开始的数据的处理策略，用户需重点关注，接口包括：

- user_flash_hook_by_fastoff，用于 AT+FASTOFF=1 快速深睡时用户数据回调接口；
- user_flash_hook_by_RB，用于由 AT+NRB/AT+NV=SAVE/AT+NATSPEED 等命令触发的软重启时用户数据回调接口；
- user_flash_hook_by_reset，用于由 AT+RESET/assert/watchdog 等触发的异常软重启后用户数据的回调接口；
- user_hook_flash_restore，用于正常上电开机后，恢复用户的数据；
- user_hook_flash_save，用于正常深睡时，保存用户的数据。

为了避免坏块，不建议用户在工作期间频繁的保存数据到 flash 中，而是建议在深睡或软重启时一次保存数据，具体接口参看 xy_flash.h。若用户产品写 Flash 的频率小于一小时每次，必须与我司 AE 人员联系，调整磨损区范围，否则会坏块。

3.6.2 用户出厂 NV 的使用建议

由于二次开发需要保存用户私有的出厂 NV，建议用户自行对 USER_FLASH_BASE 空间进行管理划分，例如划分出一个扇区 4K 空间作为用户的出厂 NV。

由于写 Flash 时发生异常断电，会造成 Flash 内容的异常，所以建议用户对重要的 flash 内容进行魔术数字和 checksum 校验等保护，具体实施参看上一章节。另外，系统在深睡及软重启时都提供了相应的 HOOK 函数，供用户进行用户 Flash 空间的读写及擦除操作，具体实施参看上一章节。

为了方便用户自行二次开发，与平台解耦，不支持用户出厂 NV 的初始值设置。即重烧版本或 FOTA 升级等场景，不会关心用户的出厂 NV 配置区域内容。要求用户自行通过代码初始化方式对私有的出厂 NV 进行赋初值。伪代码如下：

```
typedef struct
{
    int          magic;    //such as 5A5A5A5A

    char         param1;
    char         param2;
    char         padding[2];

    char         padding2[64];

    int          checksum;
}T_user_NvInfo;

VOID user_init()
{
    read user_nv_flash data; //从 flash 中读取用户 NV 数据
```

```

if(user_nv_flash data is invalid) //磨损数字+checksum 检验有效性
{
    init user nv; //对每个用户 NV 参数进行赋初值

    write user nv back to flash; //将 RAM 区的用户 NV 数值写入 flash
}
}
    
```

3.6.3 NV 使用说明及异常场景处理介绍

使用 flash 保存关机后需要保存的重要环境变量参数，即 NV。目前，平台根据 NV 改变的场景频率，提供了四块磨损区，分别为 NV_FLASH_FACTORY_BASE(出厂 NV 专用，写概率低)、NV_FLASH_DSP_VOLATILE_BASE(平台运行过程中易变 NV 区，写概率高)、NV_FLASH_DSP_NON_VOLATILE_BASE(平台运行过程中非易变 NV 区，写概率低)、NV_FLASH_NET_BASE(云通信实时信息区，写概率高)。其中，NV_FLASH_NET_BASE 为电信云、OneNET 等深睡保存注册信息用的，以便下次唤醒后直接联网，如果用户自行开发自己的云平台业务，可以直接使用该磨损区。

平台针对不同的异常场景，针对性的进行 NV 的设计，具体包括：

- PIN_RESET: 按键重启，3GPP 相关的 NV 内容保持不动，正常开机 Attach;
- WDT_RESET: 看门狗异常重启，属于严重异常，擦除所有工作态 NV，仅保持出厂 NV 有效，重启后会进行较长时间的小区搜索驻留动作;
- assert: 软件断言，属于严重异常，擦除所有工作态 NV，仅保持出厂 NV 有效，重启后会进行较长时间的小区搜索驻留动作;
- SOFT_RESET: 由软件命令触发软重启，如 AT+NRB、AT+NV=SAVE、AT+COLDNRB 三条命令，重启之前，将会保存 3GPP 相关的所有 NV，但清空平台业务相关 NV，重启后正常开机 attach;
- 恢复出厂设置: AT+RESET 命令，皆会擦除所有的 NV，包括工作态的出厂 NV，恢复到第一次烧机状态，用户的出厂配置修改被清除，会进行较长时间的小区搜索驻留动作;
- 异常断电: 即不支持断电的产品形态，发生了异常断电，将会擦除所有工作态 NV，仅保持出厂 NV 有效，会进行较长时间的小区搜索驻留动作;
- flash 操作异常: 将会擦除所有工作态 NV，仅保持出厂 NV 有效，会进行较长时间的小区搜索驻留动作;

3.6.4 OPENCPU 秒级别数据保存方案设计

对于 OPENCPU 产品，存在仅 M3 核工作，以分钟甚至秒为单位进行周期性唤醒，采集数据后保存，待达到一定的触发量，再一次性发送给远程服务器。这样的目的对于低功耗和节省流量都有很大好处。

由于周期过短，进而不能每次唤醒后将数据写入 flash，否则坏块可能大大增加。为此，平台提供了一个全局内存 g_user_vol_data，长度上限为 USER_VOL_DATA_LEN(3740 字节)的 retention memory 内存

空间，供用户来保存临时数据，参见 `user_task_volatile_demo.c` 参考代码。该段内存在深睡期间，不会断电。为了防止内存异常，提供了 checksum 检错，若发现错误，则清零。若用户需要保存更大的易变数据，则可以在 `user_hook_flash_save` 回调接口中，将 `g_user_vol_data` 保存到某个用户 flash 空间中，然后清空 `g_user_vol_data`。

由于平台默认 retention memory 用来保存自己平台的易变数据的，所以用户必须谨慎使用该功能。使用时，需要确保如下几个出厂 NV 值设置正确，即 `high_freq_data=1`；`keep_retension_mem=1`，`work_mode=1`。

对于长供电 OPENCPU 产品，需要关闭深睡模式，进而也就无需使用该方案，直接动态申请普通内存即可。

若需使用该方案，请与海凌科的 FAE 联系！

3.6.5 NV 及 flash 的调试

为了方便用户二次开发时查看 NV 及 Flash 值，平台提供了 AT 命令：`AT+TEST=READ,<addr>,<len>`，供用户查看对应的地址值。具体的 Flash 内存划分，可以查看头文件：`xy_memmap_chipB0.h`。

用户可以参考 `user_flash_demo.c` 中的代码进行开发。

4 DEMO 实例使用说明

4.1 用户任务开发指导

平台 SDK 开发环境, 提供了用户任务创建的参考代码, 参见文件 user_task_demo.c。用户仅需在 main.c 的 user_app_entry 接口中, 进行用户任务的初始化; 在 user_sys_init 中, 添加系统级初始化接口, 如 GPIO 重映射等。

需要特别指出的是, 由于芯片内部未进行周期性的世界时间同步, 建议使用 xy_set_rtc 接口进行相对时间的定时设置, 不建议设置周期性的世界时间定时。若用户需要世界时间定时功能, 可由用户自行开发世界时间同步相关代码进行世界时间的维护。



用户二次开发只能使用 userapp 下头文件中的接口、宏值等, 不得调用非 userapp 目录下的任何接口, 也不准修改非 userapp 下的任何代码。

4.2 DEMO 功能列表

功能集	源文件	使用和调试方法	备注
用户任务	user_task_demo.c	AT+NV=SET,DEMOTEST,1 AT+NRB 使用: user_app_init 中直接调用 user_task_demo_init 即可	双核启动的用户周期性 RTC 动作的参考任务代码。包括第一次上电及深睡唤醒的参考建议, 用户在 user_process 中实现产品需要做的事务, 如云通信等。双核启动的 OPENCPU 产品推荐使用。
	user_task_demo2.c	AT+NV=SET,DEMOTEST,2 AT+NRB 使用: user_app_init 中直接调用 user_task_demo2_init 即可	长供电场景下, 用户周期性 RTC 数据传输的参考代码。增加了对用户任务是否正常运行的软看门狗机制, 当未能超时喂狗, 则建议设备软重启。OPENCPU 推荐使用。
	user_task_demo3.c	AT+NV=SET,DEMOTEST,4 AT+NRB 使用: user_app_init 中直接调用 user_task_demo3_init 即可	单核启动的周期性 RTC 动作的参考任务代码, 如烟感火警报警器。包含单核数据采集及阈值满足后的动态使用 NB 等行为。用户在 user_process 中实现产品需要做的事务, 如云通信等。单核启动的 OPENCPU 推荐使用。

	user_task_volatile_demo.c	AT+NV=SET,HIGHFREQ,1 AT+NV=SET,BACKUPKEEP,1 AT+NV=SET,WORKMODE,1 AT+NV=SET,DEMOTEST,8 AT+NRB	用户高频率周期性数据采集的参考任务代码。用户高频率数据保存在 g_user_vol_data 全局中。 OPENCPU 推荐使用。如需使用,请联系海凌科的 FAE。
	user_task_mcu_demo.c		底板 MCU 控制芯片的参考任务代码,包括工作锁机制、断电模式、省电控制等参考流程。外接 MCU 用户参考使用
AT 机制 用户业务	user_task_mcu_demo.c	在 user_app_init 中调用 user_task_mcu_init 即可	底板 MCU 控制芯片的参考任务代码,包括工作锁机制、断电模式、省电控制等参考流程。外接 MCU 用户参考使用
	at_socket_ext_demo.c	在 user_app_init 中调用 at_socket_demo_init 即可	扩展 AT 参考代码,仅支持 UDP,单路。外接 MCU 用户参考使用
	at_cdp_demo.c	在 user_app_init 中调用 cdp_task_demo_init 即可	外接 MCU 用户参考使用
	at_onenet_demo.c	在 user_app_init 中调用 onenet_atcmd_task_demo_init 即可	OneNET 用户端参考代码,用户自行根据需求进行功能宏的开关,外接 MCU 用户参考使用
	at_server_demo.c	在 user_app_init 中调用 at_server_demo_entry 即可	AT 服务端参考代码,供用户实现外部 MCU 与芯片间的扩展 AT 命令信息交互,不建议使用。
	at_NFTCIMEI_demo.c	在 user_AT_Req_init 中选择性打开编译	IMEI 号的用户扩展 AT 命令示例,内部通过标准 3GPP 的 AT 命令获取 IMEI
	at_NUESTATS_demo.c	在 user_sys_init 中调用 init_NUESTATS_demo_task 接口	通过 NUESTATS 扩展 AT 命令,获取 3GPP 实时信息的解析示例,以指导用户对非标准 AT 应答结果 URC 的解析
OPENCPU 接口机制用户业务	cdp_opencpu_demo.c	在 user_app_init 中调用 cdp_opencpu_demo_init 即可	API 机制的 CDP 用户应用参考代码,流程较为复杂,建议使用简化的参考代码 cdp_opencpu_simple_demo.c
	cdp_opencpu_simple_demo.c	在 user_app_init 中创建 cdp_opencpu_simple_demo_task 任务即可	API 机制的 CDP 用户应用参考代码,OPENCPU 推荐使用,可与 user_task_demo.c 配套使用

	onenet_opencpu_demo.c	user_app_init 中调用 onenet_api_task_demo_init 即可	API 机制的 onenet 用户应用参考代码，支持多 object 多 resource。普通用户建议使用 onenet_opencpu_simple_demo.c。
	onenet_opencpu_simple_demo.c	user_app_init 中调用 cis_simple_api_task_demo_init 即可	简化版 API 机制的 onenet 用户应用参考代码，仅支持一个 object，一个 resource。OPENCPU 推荐使用，可与 user_task_demo.c 配套使用
	cloud_demo.c	电信、联通： AT+NV=SET,DEMOTEST,32 AT+NRB 中移： AT+NV=SET,DEMOTEST,64 AT+NRB	OPENCPU 的数据通信压力测试 demo，包括 OneNET 和 CDP 两个云平台，包含了各种超时容错机制
	multi_sock_demo.c	AT+NV=SET,DEMOTEST,16 AT+NRB	双路 Socket 收发线程示例，若用户有 socket 开发经验，无需参考该 demo，直接使用标准的 POSIX 套接字编程即可。
驱动相关	csp_uart_demo1.c		将 CSP4 配置成 UART 的参考代码，实现简单的数据回写，仅供用户调通 CSP，不保证功能的完整性
	csp_uart_demo2.c		使用 CSP4，配合 timeout 和 threshold，一段时间没有收到数据后，认为一包数据接收完毕，进行处理，处理过程由用户实现
	user_flash_demo.c	在 sys_hook_func.c 各个回调接口中插入调用点	深睡、唤醒、重启等对用户 flash 空间的具体实现，编程实现时请务必阅读上一章节的 Flash 使用说明，以及《海凌科 HLK-N10 API 接口说明》文档。
	spi_slave_demo.c		SPI 做 slave 的参考代码，仅实现 SPI 的接收和一些自定义命令处理的处理，具体 slave 的通信规范由用户自己实现
	spi_master_demo.c		SPI 做 master 的参考代码，仅发送，波形正常

gpio_int_demo.c		GPIO 中断配置参考代码，仅提供整体模板，产生中断需要进行的处理由用户实现
i2c_demo.c		I2C 参考代码，测试通信正常，用户调用相关接口实现所需功能
adc_demo.c		ADC 外设使用参考代码，读取外部输入的电压值
timer_demo.c		TIMER 硬定时器参考代码，实现 500ms 定时产生中断，产生中断需要进行的处理由用户实现。若用户做定时功能，建议用 RTC 定时器，见 user_task_demo.c。
rtc_demo.c	自行拷贝后，放在自己文件中编译使用	供用户参考实现每天或每周的定时行为，并提供随机值来均分用户动作时间点。
at_encode_demo.c	AT+TEST=ENCODE,<data>	mbedtls 加解密接口示例，重点关注 xy_encrypt、xy_decrypt 两个接口的使用

4.3 DEMO 开发使用说明

海凌科提供的 DEMO 是最大集功能，占用较多的 Flash 和 RAM 空间，用户使用时，必须在 targets\xinyiNBSoc_M3\Makefile\feature.mk 中，设 DEMO_SUPPORT=n，以关闭所有 DEMO。用户想参考哪个 DEMO，自行拷贝到新的文件中二次编程，不得在源文件中直接修改，否则带来代码同步和内存紧张等风险。

所有的 AT 机制用户业务参考代码，仅供用户在底板 MCU 开发时参考其 AT 流程，该代码不可运行在 M3 核。所有在 M3 核上开发的网络业务应用，只能调用海凌科提供的 API 接口，不能使用 AT 命令进行业务开发。

平台提供了硬看门狗机制，来确保软件正常调度运行。对于业务软件的异常，如无线环境很差，再如用户任务死循环等，用户需要自行设计软看门狗机制，如规定时间内未能与网络完成通信等，则触发重启等策略，具体参看 user_task_demo.c。

4.4 用户二次开发文件介绍

源文件	功能说明	备注
sys_hook_func.c	基础平台系统级 HOOK 回调，包括 flash 相关回调	该类接口在系统运行的特殊点被调用，如初始化、深睡、重启等
user_hook_func.c	用户二次开发的驱动、任务等初始	

	化点	
at_hook_func.c	AT 命令相关的 HOOK 回调函数	根据用户需求, 后续持续添加
at_ps_urc.c	URC 主动上报的处理函数集	
at_user_req.c	非 3GPP 的扩展 AT 命令, 用户有不同的需求, 该文件提出大部分用户需要修改的扩展 AT 命令供修改或增加	用户根据需要可在 user_AT_Req_init 接口中新增或删除特定 AT 命令
at_user_req1.c	用户定制的扩展 AT 命令	专用用户 AT 命令, 不需要的用户可在 user_AT_Req_init 中删除
onenet_api.c	OneNET 云 API, 支持多用户 object	一般用户不建议使用多 object, 使用 onenet_simple_api.c 即可
onenet_simple_api.c	简化的 OneNET 接口, 仅支持一个用户 object	具体使用可参看 onenet_opencpu_simple_demo.c。如果需要多个用户 object, 请联系海凌科 FAE
xy_cdp_api.c	CDP 相关 API 接口	具体使用可参看 cdp_opencpu_simple_demo.c
xy_fota.c	FOTA 相关的 API 接口	根据用户产品需要, 持续完善

版本历史

资料版本	日期	更新描述
V1.0	2020/7/18	初始版本