

AMKNxxx 系列 工控板 (模块) AT 指令表

(2022 年 11 月 1 日修订版)


版权声明

本产品使用手册包含的所有内容均受版权法的保护，未经北京中嵌凌云电子有限公司的书面授权，任何组织和个人不得以任何形式或手段对整个手册和部分内容进行复制和转载。

免责声明

本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除在其产品的销售条款和条件声明的责任之外，我司概不承担其他责任。并且我司对本产品的销售和使用不作任何明示或暗示的担保，包括对产品特定用途的适用性，适销性或对任何专利权、版权或其他知识产权的侵权责任等均不作担保。我司对文档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，我司可能随时会对产品描述和相关的功能调整或技术改进，保留修改文档中任何内容的权利，恕不另行通知。

商标声明

、AMKN 均系北京中嵌凌云电子有限公司注册商标，未经书面授权，任何人不得以任何方式使用该商标、标记。

销售及服务网络

北京

销售电话：185 0042 1002

地 址：北京市海淀区吴家场路 1 号院 2 号楼

邮 箱：sales@embedarm.com

西安

销售电话：029-6888 8268（工作日）

手 机：189 9285 2102

地 址：西安市曲江新区旺座曲江 H 座 3003 室

邮 箱：sales@embedarm.com

技术支持：

电 话：029-8877 2044（工作日）

手 机：188 0108 0298

微 信：133 9928 8868

邮 箱：embedarm@126.com

网 址：www.embedarm.com

版本变更

下面表格显示本产品使用手册在不同时期的修订版本及修订原因说明：

版本	修改内容	完成日期	修订部门
V1.00	正式发布	2020.8.29	研发部
V1.01	增加 AT+DISK, AT+FILE 指令	2020.10.28	研发部
V1.02	修改部分指令	2021.11.19	研发部
V1.03	修改部分文档	2022.6.15	研发部
V1.04	修改部分文档	2022.11.1	研发部

产品型号：

序号	类型	订货型号	备注
1	工业控制板	AMKN8600、AMKN8602G、AMKN8612、 AMKN8612G、AMKN8616G、AMKN8626、 AMKN8628、AMKN8630、AMKN407VE-BS-DK、 AMKN307VE-BG-DK、AMKN303VE-BG-DK	

特别提醒：本软件只适用于以上产品硬件，**不适用于**其它产品。

信息输出

序号	指令	属性	功能	例子说明
1	AT+INFO	输出	一般 DEBUG 信息输出	AI 初始化 OK AT+INFO=AI_Init OK!
2	AT+WARNING	输出	警告信息输出	RTC 定时中断警告 AT+WARNING=产生秒中断
3	AT+ERROR	输出	出错信息输出	AI 初始化失败 AT+ERROR=AI_Init ERROR!

只读设备信息指令

序号	指令	属性	功能	例子说明
1	AT+VER	只读	读取 AT 指令版本 (最大 5 个字符)	读取: AT+VER=? 返回: AT+VER=1.02
2	AT+MODEL	只读	读取产品型号 (最大 16 个字符)	读取: AT+MODEL=? 返回: AT+MODEL=EMB8600I
3	AT+NAME	只读	读取产品名称 (最大 32 个字符)	读取: AT+NAME=? 返回: AT+NAME=中嵌凌云工控板
4	AT+HWVER	只读	读取硬件版本 (最大 5 个字符)	读取: AT+HWVER=? 返回: AT+HWVER=1.10
5	AT+SWVER	只读	读取软件版本 (最大 5 个字符)	读取: AT+SWVER=? 返回: AT+SWVER=1.10
6	AT+LIBVER	只读	读取驱动库版本 (最大 5 个字符)	读取: AT+LIBVER=? 返回: AT+LIBVER=1.10
7	AT+BIN	只读	读取固件信息 (最大 32 个字符)	发送: AT+BIN=? 返回: AT+BIN=AMKN8600I_HW110SW110_20190101.BIN

配置和控制指令

序号	指令	属性	功能	例子说明
1	AT+RESET	配置	设备重启	重启: AT+RESET 返回: AT+RESET=OK
2	AT+SAVE	配置	保存配置数据	保存: AT+SAVE 返回: AT+SAVE=OK
3	AT+DEFAULT	配置	恢复出厂设置	设置: AT+DEFAULT 返回: AT+DEFAULT=OK
4	AT+IP	读写	设置和读取本地 IP	设置本地 IP: AT+IP=192. 168. 1. 10 返回 AT+IP=OK 读取本地 IP: AT+IP=? 返回 AT+IP=192. 168. 1. 10
5	AT+PORT	读写	设置和读取本地端口号	设置本地 PORT: AT+PORT=5000 返回 AT+PORT=OK 读取本地 PORT: AT+PORT=? 返回 AT+PORT=5000
6	AT+GWIP	读写	设置和读取网关 IP	设置网关 IP: AT+GWIP=192. 168. 1. 1 返回 AT+GWIP=OK 读取网关 IP: AT+GWIP=? 返回 AT+GWIP=192. 168. 1. 1
7	AT+DSCIP	读写	设置和读取服务器 IP 设置/读取数据服务器 1/2 IP 地址	设置: AT+DSCIP=1, 192. 168. 1. 44 返回: AT+DSCIP=OK 设置: AT+DSCIP=2, 192. 168. 1. 45 返回: AT+DSCIP=OK 读取: AT+DSCIP=? 返回: AT+DSCIP=1, 192. 168. 1. 44 AT+DSCIP=2, 192. 168. 1. 45
8	AT+DSCPORT	读写	设置/读取数据服务器 1/2/端口号	设置: AT+DSCPORT=1, 5001 返回: AT+DSCPORT=OK 设置: AT+DSCPORT=2, 5002 返回: AT+DSCPORT=OK 读取: AT+DSCPORT=? 返回: AT+DSCPORT=1, 5001 AT+DSCPORT=2, 5002

9	AT+SUBMASK	读写	设置和读取子网掩码	设置子网掩码： AT+SUBMASK=255. 255. 255. 0 返回 AT+SUBMASK=OK 读取子网掩码：AT+SUBMASK=? 返回 AT+SUBMASK=255. 255. 255. 0
10	AT+MAC	读写	设置和读取 MAC 地址	设置 MAC 地址 (16 进制)： AT+MAC=00, 01, 22, 33, 44, 55 返回 AT+MAC=OK 读取 MAC 地址 (16 进制)：AT+MAC=? 返回 AT+MAC=00, 01, 22, 33, 44, 55
11	AT+ID	读写	设置本设备唯一设备识别码 最大长度 16 字节	设置 ID：AT+ID=ZQLY001 返回 AT+ID=OK 读取 ID：AT+ID=? 返回 AT+ID=ZQLY001
12	AT+DEBUG		设置调试信息输出 ON：启动所有调试信息输出 OFF：关闭所有调试信息输出 ON, DI：启动 DI 调试信息输出 OFF, DI：关闭 DI 调试信息输出 ON, DO：启动 DO 调试信息输出 OFF, DO：关闭 DO 调试信息输出 ON, AI：启动 AI 调试信息输出 OFF, AI：关闭 AI 调试信息输出 ON, AO：启动 AO 调试信息输出 OFF, AO：关闭 AO 调试信息输出 ON, PWM：启动 PWM 调试信	设置所有调试信息输出： AT+DEBUG=ON 返回：AT+DEBUG=OK 关闭所有调试信息输出： AT+DEBUG=OFF 返回：AT+DEBUG=OK 设置 DI 调试信息输出： AT+DEBUG=DI, ON 返回：AT+DEBUG=OK 关闭 DI 调试信息输出： AT+DEBUG=DI, OFF 返回：AT+DEBUG=OK

			<p>息输出</p> <p>OFF, PWM: 关闭 PWM 调试</p> <p>信息输出</p> <p>ON, FCLK: 启动 PWM 调试</p> <p>信息输出</p> <p>OFF, FCLK: 关闭 PWM 调试</p> <p>信息输出</p> <p>ON, UART: 启动 UART 调试</p> <p>信息输出</p> <p>OFF, UART: 关闭 UART 调试</p> <p>信息输出</p> <p>ON, CAN: 启动 CAN 调试</p> <p>信息输出</p> <p>OFF, CAN: 关闭 CAN 调试</p> <p>信息输出</p> <p>ON, NET: 启动 NET 调试</p> <p>信息输出</p> <p>OFF, NET: 关闭 NET 调试</p> <p>信息输出</p> <p>ON, TIM: 启动 TIM 调试</p> <p>信息输出</p> <p>OFF, TIM: 关闭 TIM 调试</p> <p>信息输出</p> <p>ON, INFO: 启动其它调试</p> <p>信息输出</p> <p>OFF, INFO: 关闭其它调试</p> <p>信息输出</p>	
13	AT+TEST		<p>设置测试使能</p> <p>ON: 启动测试</p> <p>OFF: 关闭测试</p>	<p>启动测试:</p> <p>AT+TEST=ON</p> <p>返回: AT+TEST=OK</p> <p>关闭测试:</p> <p>AT+TEST=OFF</p> <p>返回: AT+TEST=OK</p>

读写数据指令

序号	指令	属性	功能	例子说明
1	AT+DI	只读	读取所有 DI 输入值 例子中:8 是 DI 数量, 1010010 是 DI1-DI8 的输入值	读取所有 DI: AT+DI=? 返回: AT+DI=8, 1010010
			读取 1 路 DI 输入值 例子读取 DI3 的输入值	读取 DI3: AT+DI3=? 返回: AT+DI3=1
2	AT+D0	只读	读取所有 D0 输出值 例子中:8 是 D0 数量, 11000011 是 D01-D08 输出值	读取所有 D0: AT+D0=? 返回: AT+D0=8, 11000011
			读取一路 D0 输出值 例子中读取 D04 输出值	读取 D04: AT+D04=? 返回: AT+D04=1
		控制	控制所有 D0 输出 例子中:8 是 D0 数量, 11000011 是 D01-D08 输出值	控制所有 D0: AT+D0=8, 11000011 返回: AT+D0=0K
			控制一路 D0 输出值 例子中控制 D04 输出值为 1	控制 D04: AT+D04=1 返回: AT+D04=0K
3	AT+AI	只读	读取所有 AI 输入值, 例子中:4 是 AI 数量, 560/2200/1000/2000 是 AI1-AI4 的输入值	读取: AT+AI=? 返回: AT+AI=4, 560, 2200, 1000, 2000
			读取 1 路 AI 输入值, 例子中读取 AI3 输入值	读取 AI3: AT+AI3=? 返回: AT+AI3=1000
4	AT+A0 注意: 只有在手动控制输出模式才可以用 AT 指令控制	只读	读取所有 A0 输出值, 例子中:2 是 A0 数量, 500/1000 是 A01/A02 的输出值;	读取: AT+A0=? 返回: AT+A0=2, 500, 1000
			读取 1 路 A0 输出值, 例子中, 是读取 A02 输出值;	读取: AT+A02=? 返回: AT+A02=500
		控制	控制所有 A0 输出值, 例子中:2 是 A0 数量, 500/1000 是 A01/A02 的输出值;	控制: AT+A0=2, 500, 1000 返回: AT+A0=0K
			控制 1 路 A0 输出值, 例子中, 是控制 A02 输出值;	控制: AT+A02=500 返回: AT+A02=0K

5	AT+PWM	只读	<p>读取 PWM 状态:</p> <p>AT+PWMX=?</p> <p>PWM 启动输出时返回:</p> <p>AT+PWMX=S, P1, P2, P3</p> <p>X: 表示 PWM 序号, 一般 1, 2, 3, 4</p> <p>S: 表示启动; P1: 是频率; P2: 是占空比, 范围 0 (0%)–100. 0 (100. 0%)</p> <p>P3: 表示输出脉冲个数, 如果是 0 表示持续输出不停止;</p> <p>PWM 停止输出时返回: AT+PWMX=OFF</p>	<p>读取 PWM 状态: AT+PWM2=?</p> <p>停止时返回: AT+PWM2=OFF</p> <p>正在运行时返回:</p> <p>AT+PWM2=S, 1000, 400, N</p> <p>N 表示发送脉冲个数</p> <p>表示正在执行输出状态</p>
		控制	<p>读写控制 PWM, 控制指令如下:</p> <p>启动 PWM 输出:</p> <p>AT+PWMX=S, P1, P2, P3</p> <p>X: 表示 PWM 序号, 一般 1, 2, 3, 4</p> <p>S: 表示启动; P1: 是频率; P2: 是占空比, 范围 0 (0%)–100 (100%)</p> <p>P3: 表示输出脉冲个数, 如果是 0 表示持续输出不停止;</p> <p>更改脉冲频率: AT+PWMX=F, P1</p> <p>X: 表示 PWM 序号, 一般 1, 2, 3, 4</p> <p>F: 表示更改脉冲频率; P1: 是频率;</p> <p>更改占空比: AT+PWMX=F, P1</p> <p>X: 表示 PWM 序号, 一般 1, 2, 3, 4</p> <p>R: 表示更改占空比; P1: 是占空比, 范围 0 (0%)–100 (100%)</p> <p>停止输出: AT+PWMX=OFF</p>	<p>PWM2 启动输出 10 个 1000HZ, 占空比 40%的脉冲:</p> <p>写入:</p> <p>AT+PWM2=S, 1000, 40, 10</p> <p>发回: AT+PWM2=OK</p> <p>PWM2 持续输出 2000HZ, 占空比 60%的脉冲:</p> <p>写入:</p> <p>AT+PWM2=S, 2000, 60, 0</p> <p>发回: AT+PWM2=OK</p> <p>修改脉冲频率为 5000HZ</p> <p>写入: AT+PWM2=F, 5000</p> <p>发回: AT+PWM2=OK</p> <p>修改占空比为 50%</p> <p>写入: AT+PWM2=R, 50</p> <p>发回: AT+PWM2=OK</p> <p>停止 PWM 输出:</p> <p>写入: AT+PWM2=OFF</p> <p>发回: AT+PWM2=OK</p>
		只读	<p>与 AT+PWM 配合使用</p> <p>读取方向控制状态:</p>	<p>读取: AT+PWM2=DIR, ?</p> <p>返回: AT+PWM2=DIR, 1</p>

			<p>AT+PWMX=DIR, ?</p> <p>返回: AT+PWMX=DIR, 1 或 0</p> <p>正向运转返回: AT+PWMX=DIR, 1</p> <p>反向运转返回: AT+PWMX=DIR, 0</p> <p>读取使能控制状态:</p> <p>AT+PWMX=ENA, ?</p> <p>PWM 使能返回: AT+PWMX=ENA, 1</p> <p>PWM 关闭返回: AT+PWMX=ENA, 1</p> <p>X: 表示 PWM 序号, 一般 1, 2, 3, 4</p>	<p>读取: AT+PWM2=ENA, ?</p> <p>返回: AT+PWM2=ENA, 1</p>
		控制	<p>与 AT+PWM 配合使用</p> <p>控制运转方向:</p> <p>正向运转: AT+PWMX=DIR, 1</p> <p>反向运转: AT+PWMX=DIR, 0</p> <p>返回: AT+PWMX=OK</p> <p>控制使能输出值</p> <p>使能: AT+PWM2=ENA, 1;</p> <p>关闭, AT+PWM2=ENA, 0</p> <p>返回: AT+PWMX=OK</p> <p>X: 表示 PWM 序号, 一般 1, 2, 3, 4</p>	<p>控制: AT+PWM2=DIR, 1</p> <p>返回: AT+PWM2=OK</p> <p>控制: AT+PWM2=ENA, 1</p> <p>返回: AT+PWM2=OK</p>
6	AT+FCLK	读取	<p>读取脉冲 FCLK 输入, 控制指令如下:</p> <p>AT+FCLKX=?</p> <p>X: 表示 FCLK 序号, 一般 1, 2, 3, 4</p> <p>在计数模式返回:</p> <p>AT+FCLKX=N, P1</p> <p>N 表示计数模式; P1 表示计数值</p> <p>在正交编码器模式返回:</p> <p>AT+FCLKX=D, P1, P2</p> <p>D 表示正交编码模式; P1 表示定时间隔读取的差值; P1>0 表示顺时针转动, P1<0 表示逆时针转动; P2 表示总的计数值, P2>0 表示位置为正, P2<0 表示位置为负;</p> <p>在测频模式返回:</p> <p>AT+FCLKX=F1, P1, P2, P3, ...</p>	<p>读取: AT+FCLK2=?</p> <p>计数模式返回:</p> <p>返回计数值:</p> <p>AT+FCLK2=N, 1200</p> <p>在正交编码器模式返回:</p> <p>AT+FCLK2=D, -100, -19999</p> <p>在测频模式返回:</p> <p>如果测量通道有 3 路则返回 3 条指令:</p> <p>AT+FCLK2=F1, 1000, 999, 998, 1000</p> <p>AT+FCLK2=F2, 2000, 1999, 1998, 2000</p> <p>AT+FCLK2=F3, 5000, 4999, 4998, 5000</p>

			<p>AT+FCLKX=F2, P1, P2, P3, . . .</p> <p>AT+FCLKX=F3, P1, P2, P3, . . .</p> <p>AT+FCLKX=F4, P1, P2, P3, . . .</p> <p>F1-F4: 表示测频模式通道 1-4,</p> <p>P1, P2, P3. . . 表示频率值</p> <p>在测量占空比模式</p> <p>AT+FCLKX=R, P1, P2, P3, . . .</p> <p>R 表示占空比模式</p> <p>P1, P2, P3, . . . 测量的占空比值</p>	<p>在测量占空比模式返回:</p> <p>AT+FCLK2=R, 50. 1, 50, 49. 8</p> <p>, 49. 8</p>
7	AT+UART	只读	<p>读取 UART 配置: AT+UARTX=?</p> <p>返回: AT+UARTX=BAUD, BITS, PS, STOP</p> <p>X: 1~8, 表示 UART1~UART8</p> <p>BAUD: 波特率, 115200/57600/38400</p> <p>/19200/9600/4800/2400/1200</p> <p>BITS: 数据位数, 7、8 或 9</p> <p>PS: 奇偶校验位, 0: 无校验; 1: 奇校验; 2: 偶校验</p> <p>STOP: 停止位, 1 或者 2</p>	<p>读取配置: AT+UART2=?</p> <p>返回:</p> <p>AT+UART2=115200, 8, 0, 1</p>
			<p>显示发送数据:</p> <p>AT+UARTX=TX[NUM]: DATA</p> <p>X: 1~8, 表示 UART1~UART8</p> <p>TX, 表示发送数据;</p> <p>NUM, 数据字节数, 十进制</p> <p>DATA: 具体数据, 16 进制数中间加空格分隔;</p>	<p>显示发送 16 进制数:</p> <p>AT+UART2=TX[16]: 30 31 32</p> <p>33 34 35 36 37 38 39 3A 3B</p> <p>3C 3D 3E 3F</p> <p>发送字符串数据:</p> <p>AT+UART2=TX[16]: 0123456</p> <p>789ABCDEF</p> <p>返回: AT+UART2=OK</p>
			<p>显示接收数据:</p> <p>AT+UARTX=RX[NUM]: DATA</p> <p>X: 1~8, 表示 UART1~UART8</p> <p>RX, 表示接收十六进制数据;</p> <p>NUM, 数据字节数, 十进制</p> <p>DATA: 具体数据, 十六进制数中间加空格分隔;</p>	<p>显示接收到 16 进制数据:</p> <p>AT+UART2=RX[8]: 31 32 33</p> <p>34 35 36 37 38</p> <p>接收到字符串数据:</p> <p>AT+UART2=RX[8]: 12345678</p>
		控制	<p>发送数据:</p> <p>AT+UARTX=DATA</p> <p>X: 1~8, 表示 UART1~UART8</p>	<p>发送 16 进制数:</p> <p>AT+UART2=30 31 32 33 34</p> <p>35 36 37 38 39 3A 3B 3C 3D</p>

			DATA: 具体数据, 16 进制数中间加空格分隔; 注意: 数据数量要小于 80 字节	3E 3F 发送字符串数据: AT+UART2=0123456789ABCD EF 返回: AT+UART2=OK
8	AT+CAN	只读	读取 CAN 配置: AT+CANX=? 返回: AT+CANX=P1, P2 X: 1 或者 2, 表示 CAN1 或者 CAN2 P1: 表示波特率 P2: 0, 表示标准帧, 1, 表示扩展帧	读取配置: AT+CAN2=? 返回: AT+CAN2=50000, 0
			显示发送数据: AT+CANX=TX[ID, NUM]:DATA X: 1 或者 2, 表示 CAN1 或者 CAN2 ID, CAN 数据识别 ID, 十进制 NUM, 数据字节数, 十进制 DATA: 具体数据, 十六进制数中间加空格分隔;	显示发送 8 个 16 进制数: AT+CAN1=TX[1, 8]:31 32 33 34 35 36 37 38
		控制	显示接收数据格式: AT+CANX=RX[ID, NUM]:DATA X: 1 或者 2, 表示 CAN1 或者 CAN2 ID, CAN 数据识别 ID, 十进制 NUM, 数据字节数, 十进制 DATA: 具体数据, 十六进制数中间加空格分隔;	接收到 8 个 16 进制数据: AT+CAN1=RX[1, 8]:31 32 33 34 35 36 37 38
			控制发送数据: AT+CANX=ID, DATA X: 1 或者 2, 表示 CAN1 或者 CAN2 ID, CAN 数据识别 ID, 十进制 DATA: 具体数据, 16 进制数中间加空格分隔;	向 ID 为 12 的设备发送 8 个 16 进制数: AT+CAN1=12, 31 32 33 34 35 36 37 38 返回: AT+CAN1=OK
9	AT+NET	读写	读取网络配置: AT+NET=? 返回: AT+NET=P1, P2 P1 通信类型: UDP 或 TCP P2 工作模式: CLIENT, SERVER, HTTP	读取网络配置: AT+NET=? 返回: AT+NET=TCP, SERVER
			显示发送数据: AT+NET=TX[IP, PORT, NUM]:DATA	显示发送 16 进制数: AT+NET=TX[192. 168. 1. 22,

			TX:表示发送数据; IP:对方 IP 地址;PORT:对方端口号; NUM, 数据字节数, 十进制 DATA: 具体数据, 16 进制数中间加 空格分隔;	6001, 16]:30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 显示发送 ASCII 码字符: AT+NET=TX[192. 168. 1. 22, 6001, 16]:0123456789ABCD EF
			显示接收数据: AT+NET=RX[IP, PORT, NUM]:DATA TX:表示发送数据; IP:对方 IP 地址;PORT:对方端口号; NUM, 数据字节数, 十进制 DATA: 具体数据, 十六进制数中间加 空格分隔;	显示接收到 16 进制数据: AT+NET=RX[192. 168. 1. 22, 6001, 8]:31 32 33 34 35 36 37 38 显示接收到 ASCII 码字符 数据: AT+NET=RX[192. 168. 1. 22, 6001, 8]:12345678
		控制	控制发送数据: AT+NET=ID, DATA ID: 网络通信对端的编号, 范围 1-4, DATA: 具体数据, 16 进制数中间加 空格分隔; 注意: 数据数量要小于 80 字节	发送 16 进制数: AT+NET=1, 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 发送 ASCII 码字符: AT+NET=1, 0123456789ABCD EF 返回: AT+NET=OK
10	AT+RTC	读写	设置时间日期	设置: AT+RTC=2019. 7. 1, 21:22:5 9 返回: AT+RTC=OK
			读取日期时间及毫秒	读取: AT+RTC=? 返回: AT+RTC=2019. 7. 1, 21:22:5 9, 200 200 单位是 ms
11	AT+EEPROM	只读	读取 EEPROM 状态 或测试后主动发送 EEPROM 状态	读取 EEPROM 状态: AT+EEPROM=? 返回正常: AT+EEPROM=OK, 返回错误: AT+EEPROM=ERROR

			<p>读取数据:</p> <p>AT+EEPROM=RH 或 RA+ADDR, LEN RH, 读取数据按 16 进制数显示 RA, 读取数据按字符串显示 ADDR, 读出的起始地址, 十进制 LEN, 数据字节数, 十进制; 返回数据显示: AT+EEPROM=RH 或 RA ADDR, DATA 返回数据有 DATA 部分: 具体数据, 16 进制数中间加空格分隔;</p> <p>显示写入数据: AT+EEPROM=WH 或 WA ADDR, DATA</p>	<p>读取数据 16 进制显示:</p> <p>AT+EEPROM=RH200, 10 返回 16 进制数据显示: AT+EEPROM=RH200, 00 01 02 03 04 AA BB CC DD EE</p> <p>读取数据按字符串显示:</p> <p>AT+EEPROM=RA200, 10 返回字符串数据显示: AT+EEPROM=RA200, 0123456 789</p> <p>显示写入 16 进制数据:</p> <p>AT+EEPROM=WH200, 00 01 02 03 04 AA BB CC DD EE 显示写入字符串数据: AT+EEPROM=WA200, 0123456 789</p>
		控制	<p>控制写入数据:</p> <p>AT+EEPROM=RH 或 RA ADDR, DATA ADDR: 写入地址, 十进制数; DATA: 具 体数据, 16 进制数中间加空格分隔; WH, 写入 16 进制数据 WA, 写入字符串数据</p>	<p>向地址 256 写入 16 进制数:</p> <p>AT+EEPROM=WH256, 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 返回: AT+EEPROM=OK 向地址 100 写入字符串: AT+EEPROM=WA100, 0123456 789ABCDEF 返回: AT+EEPROM=OK</p>
12	AT+SPIFLASH	只读	<p>读取 SPI FLASH 状态 或测试后主动发送 SPI FLASH 状态</p>	<p>读取 SPIFLASH 卡状态:</p> <p>AT+SPIFLASH=? 返回正常: AT+SPIFLASH=OK 返回错误: AT+SPIFLASH=ERROR</p>
			<p>读取数据:</p> <p>AT+SPIFLASH=RH 或 RA+ADDR, LEN RH, 读取数据按 16 进制数显示 RA, 读取数据按字符串显示</p>	<p>读取数据 16 进制显示:</p> <p>AT+SPIFLASH=RH200, 10 返回 16 进制数据显示: AT+SPIFLASH=RH200, 00 01</p>

			ADDR, 读出的起始地址, 十进制 LEN, 数据字节数, 十进制; 返回数据显示: AT+SPIFLASH=RH 或 RA ADDR, DATA 返回数据有 DATA 部分: 具体数据, 16 进制数中间加空格分隔; 显示写入数据: AT+SPIFLASH=WH 或 WA ADDR, DATA	02 03 04 AA BB CC DD EE 读取数据按字符串显示: AT+SPIFLASH=RA200, 10 返回字符串数据显示: AT+SPIFLASH=RA200, 01234 56789 显示写入 16 进制数据: AT+SPIFLASH=WH200, 00 01 02 03 04 AA BB CC DD EE 显示写入字符串数据: AT+SPIFLASH=WA200, 01234 56789
		控制	控制写入数据: AT+SPIFLASH=WH 或 WA+ADDR, DATA ADDR: 写入地址, 十进制数; DATA: 具 体数据, 16 进制数中间加空格分隔; WH, 写入 16 进制数据 WA, 写入字符串数据	向地址 256 写入 16 进制数: AT+SPIFLASH=WH256, 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 返回: AT+SPIFLASH=OK 向地址 100 写入字符串: AT+SPIFLASH=WA100, 01234 56789ABCDEF 返回: AT+SPIFLASH=OK
			擦除扇区: AT+SPIFLASH=EAXXX, NUM EA: 表示擦除 XXX: 擦除扇区起始扇区编号 (0~N) NUM: 擦除扇区数量	擦除扇区 10 开始的 5 个 扇区: AT+SPIFLASH=EA10, 5 返回成功: AT+SPIFLASH=OK 返回失败: AT+SPIFLASH=ERROR
13	AT+NFLA SH	只读	读取 NAND FLASH 状态 或测试后主动发送 NAND FLASH 状态	读取 NAND FLASH 状态: AT+NFLASH=? 返回正常: AT+NFLASH=OK 返回错误: AT+NFLASH=ERROR
14	AT+SRAM	只读	读取扩展 SRAM 状态 或测试后主动发送 SRAM 状态	读取扩展 SRAM 状态: AT+SRAM=?

				返回正常: AT+SRAM=OK 返回错误: AT+SRAM=ERROR
15	AT+SD	只读	读取 SD 卡状态 或测试后主动发送 SD 卡状态	读取 SD 卡状态: AT+SD=? 返回正常: AT+SD=OK 返回错误: AT+SD=ERROR
16	AT+UDISK	只读	读取 U 盘状态 或测试后主动发送 U 盘状态	读取 U 盘状态: AT+UDISK=? 返回正常: AT+UDISK=OK 返回错误: AT+UDISK=ERROR
17	AT+USB	只读	读取 USB 设备状态 或测试后主动发送 USB 设备状态	读取 USB 设备状态: AT+USB=? 返回正常: AT+USB=OK 返回错误: AT+USB=ERROR
18	AT+FILE	读写	打开, 关闭, 删除, 同步文件 打开文件: AT+FILE=X, OPEN, embedarm.dat 同步文件: AT+FILE=X, SYNC 关闭文件: AT+FILE=X, CLOSE 删除文件: AT+FILE=X, DEL, embedarm.dat X: 0, 1, 2, 3 代表盘符, 0 表示 SPI FLASH; 1, 表示 SD 卡; 2, 表示 U 盘; 3, 表示 NAND FLASH; 注意: 文件名只支持: 8.3 即 8 字符 文件名. 3 字节扩展名 如果打开文件不存在, 会自动创建该 文件	打开文件: AT+FILE=0, OPEN, embedarm. dat 返回正确: AT+FILE=OK 返回失败: AT+FILE=ERROR 关闭文件: AT+FILE=0, CLOSE 返回正确: AT+FILE=OK 返回失败: AT+FILE=ERROR 同步文件: AT+FILE=0, SYNC 返回正确: AT+FILE=OK 返回失败: AT+FILE=ERROR 删除文件: AT+FILE=0, DEL, embedarm. dat 返回正确: AT+FILE=OK 返回失败: AT+FILE=ERROR
			读取打开的文件信息 AT+FILE=X, ? 返回: AT+FILE=0, embedarm.dat, size 盘符, 文件名, 文件大小	AT+FILE=0, ? 返回错误: AT+FILE=ERROR 表示无文件或文件未打开 返回正确: AT+FILE=0, embedarm.dat, 100
			列举当前磁盘根目录所有文件及文	列举根目录所有文件及文

			<p>件夹: AT+FILE=X, LIST</p> <p>列举当前磁盘 SRC 文件夹下所有文件及文件夹: AT+FILE=X, LIST, SRC</p> <p>返回:</p> <p>AT+FILE=盘符, 文件或文件夹名, 属性, 大小, 文件最后被修改日期时间; 其中属性: A, 可读写; D, 文件夹; H, 隐藏文件; S, 系统文件; R, 只读文件;</p>	<p>件夹</p> <p>AT+FILE=0, LIST 返回:</p> <p>AT+FILE=0, A001. txt, A, 23 45, 2020-10-28, 12:11:33</p> <p>AT+FILE=0, SRC, D, 0, 2020-10-29, 12:11:45</p> <p>列举 SRC 文件夹下所有文件及文件夹</p> <p>AT+FILE=0, LIST, SRC 返回:</p> <p>AT+FILE=0, A002. txt, A, 23 45, 2020-10-28, 12:11:33</p>
			<p>写入 ASCII 字符或 16 进制数据:</p> <p>AT+FILE=X, WA 或 WH+ADDR, DATA</p> <p>X(0, 1, 2, 3) 代表盘符</p> <p>WH, 写入 16 进制数据</p> <p>WA, 写入字符串数据</p> <p>ADDR, 写入的起始地址, 十进制数字; 如果写入地址处空: 表示从文件结尾处写入数据</p> <p>DATA: 具体数据, 16 进制数中间加空格分隔;</p>	<p>地址 100 开始写入 10 个 ASCII 字符数据:</p> <p>AT+FILE=0, WA100, 0123456789</p> <p>返回: AT+FILE=OK</p> <p>地址 200 开始写入 10 个 16 进制数据:</p> <p>AT+FILE=0, WH200, 00 01 02 03 04 AA BB CC DD EE</p> <p>返回: AT+FILE=OK</p>
			<p>读取 ASCII 字符或 16 进制数据:</p> <p>AT+FILE=X, RA 或 RH+ADDR, LEN</p> <p>RH, 表示读出 16 进制数据;</p> <p>RA, 表示读出 ASCII 码字符数据;</p> <p>ADDR, 读出的起始地址, 十进制</p> <p>LEN, 数据字节数, 十进制数字, 如果长度大于文件大小则读取全部文件;</p> <p>返回数据有 DATA 部分: 具体数据, 16 进制数中间加空格分隔;</p>	<p>读取: AT+FILE=0, RA100, 10</p> <p>返回:</p> <p>AT+FILE=0, RA200, 0123456789</p> <p>读取: AT+FILE=0, RH200, 10</p> <p>返回: AT+FILE=0, RH200, 00 01 02 03 04 AA BB CC DD EE</p>
19	AT+SW	只读	<p>读取拨码开关或拨码开关有拨动主动上传:</p> <p>例子中返回 8 位拨码开关值</p>	<p>读取: AT+SW=?</p> <p>返回: AT+SW=00001111</p> <p>或主动上传:</p>

			从左到右依次是 S1-S8 如果单独一位拨码开关变化会返回：	AT+SW=10001111
20	AT+KEY	主动发送	AT+KEYX=P1, P2 X: 按键编号, 1-32 P1=1, 按键按下; P1=0, 按键抬起; P2: 按键按下持续时间, 单位 ms	按键 KEY2 按下: 主动上传: AT+KEY2=1, 20 20 表示: 按键已经按下 20ms 按键 KEY2 抬起: 主动上传: AT+KEY2=0, 500 500 表示: 按键按下一共 500ms
21	AT+ALARM	只写	控制板上蜂鸣器	蜂鸣器响 100ms AT+ALARM=1, 100 蜂鸣器关闭 AT+ALARM=0
22	AT+DISK	控制	磁盘使能: AT+DISK=0, 1 磁盘关闭: AT+DISK=0, 0 格式化磁盘: AT+DISK=0, FORMAT 0 (1, 2, 3) 代表盘符, 0 表示 SPI FLASH; 1, 表示 SD 卡; 2, 表示 U 盘; 3, 表示 NAND FLASH; 注意: 在操作磁盘其它操作或者 AT+FILE 指令前必须先使能相应的磁盘 读取 SPIFLASH 储存空间信息 AT+DISK=0, ? 返回: AT+DISK=0, SIZE, FREE SIZE, 总空间, 单位 KB FREE, 空闲空间, 单位 KB	格式化 SPI FLASH: AT+DISK=0, FORMAT 返回正确: AT+DISK=OK 返回失败: AT+DISK=ERROR 读取磁盘信息 AT+DISK=0, ? 返回正确: AT+DISK=0, 8000, 3200 返回错误: AT+DISK=ERROR

- 注: 1. 每条指令以回车换行<CR><LF> (0x0D 0x0A) 为结束标志
2. 任何命令中间**不允许**加空格;
3. 所有命令字符**必须**大写;