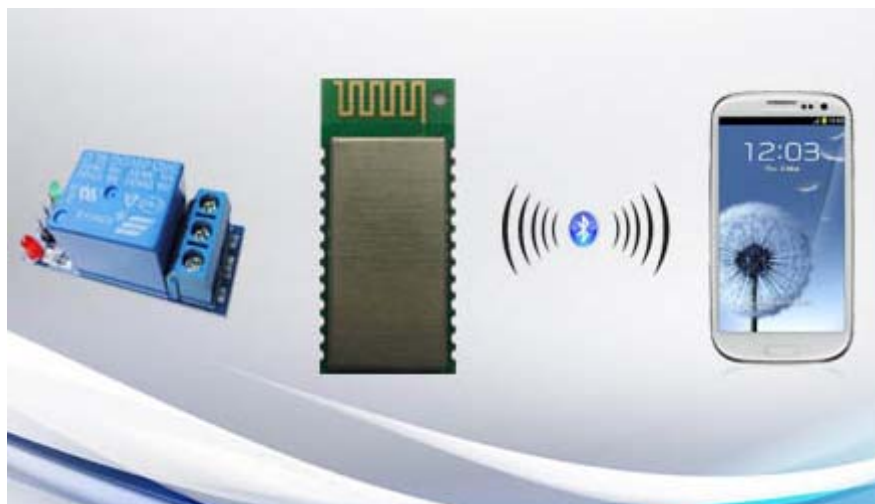




XM-15 蓝牙串口模块继电器控制接口



一、简介

重庆翔码电子推出的 XM-15 蓝牙模块，具有远程无线控制功能，可以通过手机（或者其他蓝牙设备）**控制继电器**，从而**控制电灯的开关**，也可以**控制电灯的亮度**（PWM 输出），或者控制门锁等。模块具有 **PWM 输出**（波形）和 **IO 口输出**（高低电平）两种功能。如果有特殊需要，本公司还可以提供定制开发服务，以满足不同客户的需求。



二、IO 口说明

引脚编号.	引脚名称		引脚名称	引脚编号.
1	UART_TX		PIO11	34
2	UART_RX		PIO10	33
3	UART_CTS		PIO9	32
4	UART_RTS		PIO8	31
5	PCM_CLK		PIO7	30
6	PCM_OUT		PIO6	29
7	PCM_IN		PIO5	28
8	PCM_SYNC		PIO4	27
9	AIO0		PIO3	26
10	AIO1		PIO2	25
11	RESETB		PIO1	24
12	VCC		PIO0	23
13	GND		GND	22
14~21	N/A			

XM-15 蓝牙模块有 15 个 IO 口可供控制使用(如果需要,最多还可以扩展 4 个), PIO4、PIO5 具有 **PWM 输出功能**, 其他口作为普通 IO 口, 可以**输出高低电平**。所有的 IO 口都可以在手机上通过指令无线控制蓝牙模块的 PIO 输出。

模块默认占用了 PIO1 作为工作状态指示, 未连接时闪烁, 连接后常亮。
PIO3 作为清除匹配信息的按键, 用于清除记忆的匹配信息。

PIO 编号	对应引脚编号	对应引脚名称	备注
0	23	PIO0	可控制继电器
1	24	PIO1	连接指示灯
2	25	PIO2	可控制继电器
3	26	PIO3	连接清除按键, AT+EBTN=0,0 之后, 也可以连接继电器
4	27	PIO4	可控制继电器
5	28	PIO5	可控制继电器
6	29	PIO6	可控制继电器
7	30	PIO7	可控制继电器
8	31	PIO8	可控制继电器
9	32	PIO9	可控制继电器
10	33	PIO10	可控制继电器



11	34	PIO11	连接匹配按键， AT+EBTN 禁用匹配 按键(默认就是禁用) 可以连接继电器
12	9	AIO0	可控制继电器
13	10	AIO1	可控制继电器
14	8	PCM_SYNC	可控制继电器

三、手机控制说明

可以通过 AT 指令 (AT+TYPE=1)，设置模块为远程控制模式，这样，手机连接上之后，自动进入控制模块（命令模式），然后手机端可以发送 AT 指令控制 PIO 口输出高低电平或者 PWM 波形。当需要进行串口数据透传时（手机发送的数据，通过蓝牙模块物理串口原样输出，模块从物理串口接收的数据，原样发送给手机），可以在命令模式下，发送 AT+EXIT 指令（要加\r\n），退出命令模式，进入数据透传模式。

如果参数 AT+TYPE=0（模块出厂默认值），则手机连接之后，会进入数据透传模式，要进行控制时，在手机端，发送“\$!CMDMODE!\$”（没有引号，没有回车换行），模块收到这串字符之后，会自动进入控制模式（命令模式），然后就和 AT+TYPE=1 相同的控制方式了。

四、AT 指令

所有指令都必须在命令模式下才可以发送，否则不会有任何响应。所有 AT 指令都以回车换行符结束，就是\r\n (0x0D 0x0A)，返回的应答也以回车换行符结束。

串口默认参数：9600，N，8，1，无硬件流控。

1、测试指令

指令	响应	参数
AT	OK	无

2、查询/设置设备类型

指令	响应	参数
AT+TYPE=<nType>	OK	nType: 设备类型 0 普通串口透传 1 远程控制+串口透传 默认: 0
AT+TYPE?	+TYPE:<nType> OK	

设备类型

- 0 普通串口透传，蓝牙连接后自动进入透传模式，手机等其他蓝牙设备，给模块发送的数据，原封不动的从模块的串口输出，反之亦然。



- 1 远程控制+串口透传， 蓝牙连接后自动进入远程控制模式， 对方蓝牙可以直接发送 AT 指令控制模块， 对方蓝牙发送 AT+EXIT 指令后退出远程控制模式， 进入串口透传模式。 在串口透传模式下， 可以发送 “\$!CMDMODE!\$”（没有回车换行）， 重新让模块进入远程控制模式。

3、 设置 PIO 单端口输出

指令	响应	参数
AT+PIO=<nPio>, <level>	OK	nPio: PIO 端口序号（十进制数） level: PIO 端口输出电平 0——低电平 1——高电平

XM-15 蓝牙模块为用户提供 PIO 端口资源：PIO2~PIO10， 用户可用来扩展输入、输出端口。

举例说明：

- 1)、PIO10 端口输出高电平
AT+PIO=10,1
OK
- 2)、PIO10 端口输出低电平
AT+PIO=10,0
OK

4、 设置 PIO 单端口输出并记忆

指令	响应	参数
AT+PIOS=<nPio>, <level>	OK	nPio: PIO 端口序号（十进制数） level: PIO 端口输出电平 0——低电平 1——高电平

此指令与 AT+PIO 的不同在于， 输出状态后， 会将状态记录到 Flash， 模块断电重启后， 会自动恢复最后记忆的状态。

5、 清除记忆的 PIO 输出状态

指令	响应	参数
AT+PIOCS=<nPioMask>	OK	nPioMask: 需要清除输出状 PIO 掩码。（十六进制）

此指令用于清除 AT+PIOS 指令记忆的状态。

nPioMask: 需要清除输出状态 PIO 掩码， 每一位代表一个 PIO， 1 代表相应的 PIO 需要清除， 0 代表相应的 PIO 保持之前的状态， 最低位代表 PIO0， 比如清除 PIO5、PIO7 的输出状态， 则 nPioMask=(1<<5)|(1<<7)=A0（十六进制）。

6、 设置 PIO 多端口输出

指令	响应	参数
		nPioMask: PIO掩码值（十六进



AT+MPIO=<nPioMask> , <nPioValue>	OK	制) nPioValue: PIO 端口输出状态, 每一位代表一个 PIO, 最低位代表 PIO0 (十六进制)
----------------------------------	----	---

控制模块多个端口输出高低电平

nPioMask: 要控制哪些 PIO, 每一位代表一个 PIO, 最低位代表 PIO0, 比如要控制 PIO5、PIO7, 则 nPioMask=(1<<5)|(1<<7)=A0 (十六进制数)

nPioValue: PIO 口输出的高低电平, 1 代表高电平, 0 代表低电平, 每一位代表一个 PIO, 最低位代表 PIO0, 比如 PIO5 输出高电平、PIO7 输出低电平, 则 nPioValue=(1<<5)=20 (十六进制数)

7、设置 PIO 多端口输出并记忆

指令	响应	参数
AT+MPIOS=<nPioMask> , <nPioValue>	OK	nPioMask: PIO掩码值 (十六进制) nPioValue: PIO 端口输出状态, 每一位代表一个 PIO, 最低位代表 PIO0 (十六进制)

此指令与 AT+MPIO 的不同在于, 输出状态后, 会将状态记录到 Flash, 模块断电重启后, 会自动恢复最后记忆的状态。

8、查询 PIO 当前输出的状态

指令	响应	参数
AT+MPIO?	+MPIO:<nPioState> OK	nPioState: PIO 端口输出状态, 每一位代表一个 PIO, 最低位代表 PIO0 (十六进制)

nPioState: 当前 PIO 口输出的是高电平还是低电平, 1 代表高电平, 0 代表低电平, 每一位代表一个 PIO, 最低位代表 PIO0, 比如 00A0, 就表示 PIO5 输出高电平、PIO7 输出高电平。

AT+MPIO?
+MPIO:00A0
OK

9、输出 PWM 波形

指令	响应	参数
AT+PWM=<nPio> , <nFreqHZ>, <nHighValue>, <nTotalValue>	OK	nPio: PIO 端口序号 (十进制数) nFreqHZ: PWM 波形的频率 (十进制) nHighValue: PWM 波形高电平时间 nTotalValue: PWM 波形总时间



此指令控制蓝牙模块的 PIO 口输出 PWM 波形。

nPio: 蓝牙模块的 PIO 口序号， 我们只支持 PIO4 和 PIO5 输出 PWM 波形， 所以这里只能选择 4 或者 5

nFreqHZ: PWM 波形的频率， 比如输出 1K Hz 的波形， 则取值为 1000

nHighValue, nTotalValue: 这两个值决定了 PWM 波形的占空比， 如取值 10 和 100， 则高电平占 10%的时间， 比如 128 和 1000， 则高电平的时间比例为 12.8%， 为简单起见， nTotalValue 可以固定取值 1000， 调整 nHighValue 来调整占空比就可以了。

10、输出 PWM 波形并记忆

指令	响应	参数
AT+PWMS=<nPio> , <nFreqHZ>, <nHighValue>, <nTotalValue>	OK	nPio: PIO 端口序号（十进制数） nFreqHZ: PWM 波形的频率（十进制） nHighValue: PWM 波形高电平时间 nTotalValue: PWM 波形总时间

此指令与 AT+PWM 的区别在于， 蓝牙模块会输出 PWM 的同时， 记忆 PWM 输出状态， 上电复位后后自动恢复最后设置的 PWM 输出。

11、清除记忆的 PWM 输出状态

指令	响应	参数
AT+PWMCS=<nPioMask>	OK	nPioMask: 需要清除 PWM 输出状态 PIO 掩码。

此指令用于清除 AT+PWMS 指令记忆的状态。

nPioMask: 需要清除 PWM 输出状态 PIO 掩码， 每一位代表一个 PIO， 1 代表相应的 PIO 需要清除， 0 代表相应的 PIO 保持之前的状态， 最低位代表 PIO0， 由于只有 PIO4 和 PIO5 具有 PWM 功能， 所以此处只有第 4 位和第 5 位有效， 其他位都无效， 比如清除 PIO5、PIO7 的输出状态， 则 nPioMask=(1<<5) | (1<<7)=A0（十六进制）。

说明： 模块还具有很多与控制不直接相关的 AT 指令（比如修改波特率， 修改名称， 修改密码等）， 请查看模块的完整 AT 指令说明文档。



联系方式:

重庆翔码电子科技有限公司

地址: 重庆市沙坪坝区华宇金沙港湾5号

邮编: 400030

网址: www.xiangma.cc

电话: 023-65007075

传真: 023-65007075

手机: (0)13098673616

QQ: 65957667

Email: sales@xiangma.cc