

KNX 核心板说明书

H8C09



(可单独提供协议处理芯片)

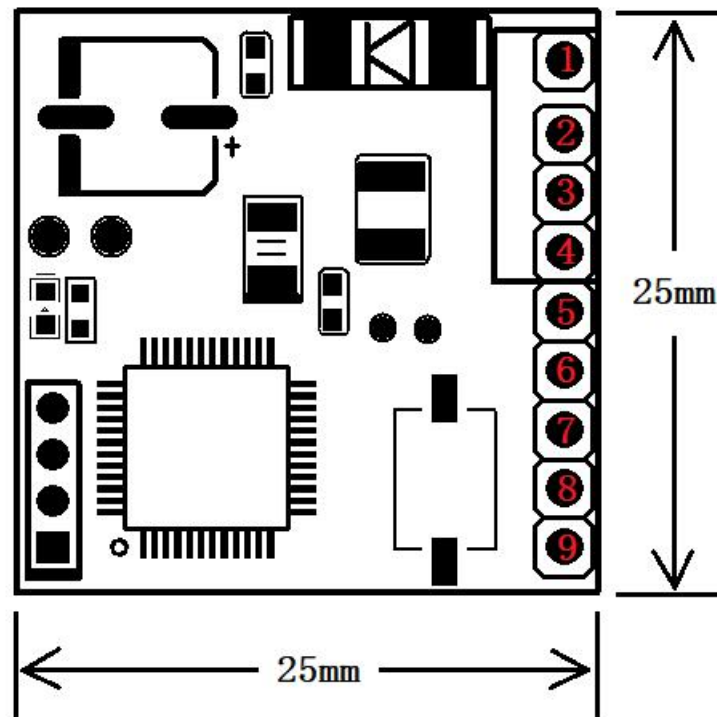
目录

1.概述.....	3
2.硬件说明.....	3
3.串口设置.....	4
4.报文格式.....	4
5.从外部发到核心板的命令.....	5
5.1 向 KNX 总线读组地址 (0x00)	5
5.2 向 KNX 总线写组地址 (0x01)	5
5.3 读通讯对象 (0x02)	5
5.4 写通讯对象 (0x03)	6
5.5 向 KNX 总线发原始数据 (0x04)	6
5.6 获取参数 (0x05)	7
5.7 串口波特率 (0x06)	7
5.8 物理地址 (0x07)	7
5.9 制造商 ID (0x08)	7
5.10 激活 (0x09)	8
5.11 发送间隔 (0x0A)	8
5.12 重发设置 (0x0B)	8
5.13 重启核心板 (0x0C)	8
5.14 编程状态 (0x0D)	8
5.15 卸载 (0x0E)	9
5.16 内部更新对象 (0x0F)	9
5.17 内部更新连续的对象 (0x10)	10
5.18 核心板应答.....	10
6.从核心板发到外部的命令.....	11
6.1 来自 KNX 总线的读组地址的回复 (0x80)	11
6.2 来自 KNX 总线的写组地址 (0x81)	11
6.3 通讯对象值更新 (0x82)	11
6.4 来自 KNX 总线的原始数据 (0x84)	12
6.5 参数 (0x85)	12
6.6 物理地址 (0x87)	12
6.7 制造商 ID (0x88)	12
6.8 核心板重启 (0x8C)	12
6.9 编程状态 (0x8D)	13
6.10 外部应答.....	13
7.命令表.....	14
8.ETS 参数.....	15
9.通讯对象.....	16
10.ETS 数据库制作.....	17
11.拿到板子了不知道怎么开始?	18

1.概述

本模块内置 KNX 协议，使用者不必深入了解协议本身就可以通过串口命令调用各种功能进行处理从而快速搭建自己的 KNX 设备。随板提供了一个通用版的数据库，开启了多种参数和通讯对象，也可以自己做数据库。

2.硬件说明



引脚编号	说明
1	KNX 正极, 24V~30V
2	KNX 正极, 24V~30V
3	GND
4	GND
5	3.3V, 核心板输出 3.3V 电压, 可对外部芯片供电 (<20mA)
6	外接编程按键输入端, 低电平有效(板内已上拉到 3.3V)
7	外接编程灯输出端, 低电平点亮(3.3V TTL)
8	串口命令接收端(3.3V TTL)
9	串口命令发送端(3.3V TTL)

注：KNX 输入端可外接一个共模电感以提高抗干扰能力。

3.串口设置

波特率：9600、19200 或 115200，默认 115200

校验位：无

停止位：1 位

4.报文格式

A.命令报文

命令报文长度为 3~64 字节。

命令类型（1 字节）+ 数据长度（1 字节）+ 数据（N 字节）+ 校验（1 字节）

命令类型：bit7 - 数据的方向。0 - 外部到核心板，1 - 核心板向外部。

Bit6~bit0 - 具体的命令，详见下文。

数据长度：数据部分的总字节数，最大为 61 字节。

数据：数据的具体含义与命令类型有关，详见下文。

校验：计算从命令类型到最后一个数据的 XOR8 校验。

命令报文字节间的间隔不大于一个字节的长度。

B.应答报文

应答报文为单一字节，用于在接收到命令报文后，应答发送方。当命令报文发出后，在规定的时间内如果未收到应答报文，则会触发重发操作。

5.从外部发到核心板的命令

5.1 向 KNX 总线读组地址（0x00）

$$00 + 03 + \text{Add_H} + \text{Add_M} + \text{Add_L} + \text{XOR8}$$

Add_H/Add_M/Add_L 分别表示组地址的高、中、低位（下同），例如读组地址 1/2/3 的报文为：00 03 01 02 03 03

当三个字节都为 0xFF 时，将关闭 0x80 功能；当收到合法的 0x00 命令报文时开启 0x80 功能。

5.2 向 KNX 总线写组地址（0x01）

$$01 + L_data + \text{Add_H} + \text{Add_M} + \text{Add_L} + L_value + V1...Vn + \text{XOR8}$$

L_data 即数据长度，表示后面的数据有多少个字节；

Add_H/Add_M/Add_L 分别表示组地址的高、中、低位；

L_value 表示组地址的值的长度，单位是 bit；

V1...Vn 表示组地址的值。

例，

向组地址 1/2/3 写开关量 1： 01 05 01 02 03 01 01 04

向组地址 1/1/7 写场景号 9： 01 05 01 01 07 08 09 02

向组地址 6/1/8 写温度 24 度：01 06 06 01 08 10 0C B0 A4

当三个字节都为 0xFF 时，将关闭 0x81 功能；收到合法的 0x01 命令报文时开启 0x81 功能。

5.3 读通讯对象（0x02）

$$02 + 01 + \text{ObjNum} + \text{XOR8}$$

ObjNum 表示通讯对象号，例如读通讯对象 **08** 的报文为：02 01 **08** 0B

在使用此命令前，应先开启对应的通讯对象，并绑定到一个组地址。执行此命令时，会在 KNX 总线上产生一个读组地址的报文，该地址为通讯对象绑定的首地址。

5.4 写通讯对象（0x03）

$$03 + L_data + ObjNum + L_obj + V1...Vn + XOR8$$

L_data 即数据长度，表示后面的数据有多少个字节；

ObjNum 表示通讯对象号；

L_obj 表示通讯对象的数据长度，单位是 bit；

V1...Vn 表示通讯对象的值。按数据长度从低位到高位取值。

在使用此命令前，应先开启对应的通讯对象，并绑定到一个组地址。执行此命令时，会在 KNX 总线上产生一个写组地址的报文，该地址为通讯对象绑定的首地址。

例，

向通讯对象 **6** 写开关量 **1**： 03 03 **06** 01 **01** 06

向通讯对象 **100** 写场景号 **9**： 03 03 **64** 08 **09** 65

向通讯对象 **200** 写温度 **24** 度： 03 04 **C8** 10 **0C B0** 63

5.5 向 KNX 总线发原始数据（0x04）

$$04 + L_data + D1...Dn + XOR8$$

L_data 即数据长度，表示后面的数据有多少个字节；

D1...Dn 表示要发送的完整的 KNX 报文。

当原始数据转发功能开启时，除了可识别成串口命令的 KNX 总线上的报文，都会把原始 KNX 报文从核心板串口转发出来；外部可以直接经核心板串口将 KNX 报文发送到 KNX 总线。

当要关闭原始数据转发功能时，向核心板发送：**04 05 70 70 83 74 6F 99**。转发功能关闭后，KNX 总线上的报文不会从核心板的串口转发出来。当外部通过此命令向核心板发送一条完整的 KNX 报文时，转发功能自动开启。

5.6 获取参数 (0x05)

$$05 + 04 + \text{AddStart_H} + \text{AddStart_L} + \text{AddEnd_H} + \text{AddEnd_L} + \text{XOR8}$$

AddStart_H/AddStart_L 组成 2 字节的首地址，AddEnd_H/AddEnd 组成 2 字节的尾地址。两个地址的取值范围是 0x3B00~0x3FFF，同时首地址及[尾地址+1]必须为 32 的倍数，且尾地址大于首地址。核心板将通过命令 0x85 返回参数，每条报文中包含固定的 32 字节参数。

5.7 串口波特率 (0x06)

$$06 + 01 + D1 + \text{XOR8}$$

D1 — 00，将波特率设为 9600；

— 01，将波特率设为 19200；

— 02，将波特率设为 115200（默认）。

5.8 物理地址 (0x07)

$$07 + 03 + \text{Add_H} + \text{Add_M} + \text{Add_L} + \text{XOR8}$$

Add_H/Add_M/Add_L 分别表示物理地址的高、中、低位，如果是合法的物理地址则将 KNX 设备的物理地址改为该值；否则视为读取物理地址，核心板将以命令 0x87 返回物理地址。（第 1 或第 2 位大于 0x0F 则为非法地址）

5.9 制造商 ID (0x08)

$$08 + 02 + \text{ID_H} + \text{ID_L} + \text{XOR8}$$

ID_H/ID_L 组成 2 字节表示制造商 ID，有效值 0001~FFFF；当为 0000 时视为读取 ID，核心板将以命令 0x88 返回 ID。

5.10 激活 (0x09)

$09 + 08 + A1...A8 + XOR8$

A1...A8 — 8 字节激活码。

出厂已激活，用户不用管。

5.11 发送间隔 (0x0A)

$0A + 02 + ThTl + XOR8$

ThTl — 表示核心板串口发送数据的最小间隔，2 字节，单位 ms，默认 20ms。

5.12 重发设置 (0x0B)

$0B + 02 + Re + To + XOR8$

Re — 重发次数，0~3，默认 3；核心板发出命令后，超时未收到应答报文的重发次数。

To — 超时时间，单位 ms，默认 10ms。核心板发出命令后，超过此时间未收到应答报文会触发重发。

5.13 重启核心板 (0x0C)

$0C + 05 + 70\ 70\ 83\ 74\ 6F + 91$
重启核心板。

5.14 编程状态 (0x0D)

$0D + 01 + Data + XOR8$

Data — 00，使核心板退出编程状态；

— 01，使核心板进入编程状态；

— 非 00 且非 01，读取编程状态，核心板将以命令 0x8D 返回编程状态。

可应用于板外编程按钮触发编程状态。

5.15 卸载 (0x0E)

$0E + 01 + \text{Type} + \text{XOR8}$

Type — 01 卸载应用

— 02 卸载应用和物理地址

卸载物理地址后，地址改成 15.15.255。卸载应用后，删除所有绑定的组地址，ETS 参数恢复默认。

5.16 内部更新对象 (0x0F)

$0F + L_data + \text{ObjNum1} + L_obj1 + \text{Value1}[\dots] + \text{ObjNum2} + L_obj2 +$

$\text{Value2}[\dots] + \dots + \text{ObjNumN} + L_objN + \text{ValueN}[\dots] + \text{XOR8}$

L_data 即数据长度，表示后面的数据有多少个字节；

ObjNumN 表示通讯对象 N 的对象号；

L_objN 表示通讯对象 N 的数据长度，单位是 bit；

ValueN[...]表示对象的值。

使用此命令，可更新核心板上的对象，而不发送到总线上。主要应用于对象需要被其他设备读取时，比如本机的温度传感器数据。

注：可一次更新多个对象，但报文长度不要超过 64 字节。

5.17 内部更新连续的对象（0x10）

$10 + L_data + \text{起始对象 } N + \text{对象 } N \text{ 的值} + \text{对象}(N+1)\text{的值} + \dots + \text{对象}(N+n)\text{的值} + \text{XOR8}$

[0x10]命令同 5.16 的[0x0F]命令都是用来更新核心板上的对象，而不发送到总线上。如果更新的对象比较多，对象号又是连续的，则[0x10]命令更高效。

5.18 核心板应答

核心板收到一组报文后，先校验其正确性，然后向外部发出单个字节的应答信号：

0xAA — 数据正确，命令可执行；

0x99 — 数据有错，命令不可执行。

应答报文不重发。

6.从核心板发到外部的命令

6.1 来自 KNX 总线的读组地址的回复 (0x80)

$$80 + L_data + Add_H + Add_M + Add_L + L_value + V1...Vn + XOR8$$

L_data 即数据长度，表示后面的数据有多少个字节；

Add_H/Add_M/Add_L 分别表示组地址的高、中、低位；

L_value 表示组地址的值的长度，单位是 bit；

V1...Vn 表示组地址的值。

注意，转发不限于本机的组地址，收到此类报文时不一定是自己所想读取的那个组地址。默认关闭。

6.2 来自 KNX 总线的写组地址 (0x81)

$$81 + L_data + Add_H + Add_M + Add_L + L_value + V1...Vn + XOR8$$

L_data 即数据长度，表示后面的数据有多少个字节；

Add_H/Add_M/Add_L 分别表示组地址的高、中、低位；

L_value 表示组地址的值的长度，单位是 bit；

V1...Vn 表示组地址的值。

默认关闭。

6.3 通讯对象值更新 (0x82)

$$82 + L_data + ObjNum + L_obj + V1...Vn + XOR8$$

L_data 即数据长度，表示后面的数据有多少个字节；

ObjNum 表示通讯对象号；

L_obj 表示通讯对象的数据长度，单位是 bit；

V1...Vn 表示对象的值。

6.4 来自 KNX 总线的原始数据 (0x84)

$$84 + L_data + D1...Dn + XOR8$$

L_data 即数据长度，表示后面的数据有多少个字节；

D1...Dn 表示要发送的完整的 KNX 报文。

注：要使用此功能需先开启原始数据转发功能；另外，来自总线的错误报文不进行转发。当原始数据为可识别命令报文时，同时发出原始数据报文和命令报文。

6.5 参数 (0x85)

$$85 + 22 + AhA1 + D1...D32 + XOR8$$

AhA1 — 这组参数的首地址，数据部分的参数的地址是连续的，具体地址表示在 ETS 数据库中的地址。根据首地址可确定参数类型。

D1...D32 — 参数数据，固定为 32 个字节。即 1bit 和 1byte 可放 32 个，2byte 可放 16 个，4byte 可放 8 个。具体的参数地址见后文。

6.6 物理地址 (0x87)

$$87 + 03 + Add_H + Add_M + Add_L + XOR8$$

Add_H/Add_M/Add_L 分别表示物理地址的高、中、低位。核心板发出当前物理地址。

6.7 制造商 ID (0x88)

$$88 + 02 + ID_H + ID_L + XOR8$$

核心板发出当前制造商 ID，ID=(ID_H<<8)+ID_L。

6.8 核心板重启 (0x8C)

$$8C + 11 + 70\ 70\ 83\ 74\ 6F + S1...S7 + VhVl + FUN + AT + XOR8$$

70 70 83 74 6F — 答疑 QQ 号 707083746；

S1...S7 — 8 字节本机串号；

VhVl — 2 字节软件版本号；

FUN — 1 字节功能开启标志，用一位代表一个功能，为 1 时开启，为 0 时关闭，如下表：

FUN							
7	6	5	4	3	2	1	0
					组 地 址 写	组 地 址 回复	原 始 数 据转发

AT — 激活状态。当为 FF 时表示已激活，否则为还可启动的次数。

核心板重启后，会首先发出此命令。

6.9 编程状态（0x8D）

$$8D + 01 + D1 + XOR8$$

D1 — 00，退出编程状态；

— 01，进入编程状态。

当核心板编程状态发生变化或被读取时，发出此命令。可应用于触发板外编程灯。

6.10 外部应答

接收到核心板最后一个字节后，在超时时间内应该给核心板一个应答报文，否则核心板会重发。

0x2A — 数据正确；

0x19 — 数据有错。

应答报文不重发。

7.命令表

方向	命令类型	描述	备注
外部到核心板	00	读组地址	
	01	写组地址	
	02	读通讯对象	
	03	写通讯对象	
	04	原始数据	
	05	获取参数	
	06	串口波特率	
	07	物理地址	
	08	制造商 ID	
	09	激活	
	0A	发送间隔	
	0B	重发设置	
	0C	重启	
	0D	编程状态	
	0F	内部更新对象	
	10	内部更新连续对象	
核心板到外部	80	读组地址回复	
	81	写组地址	
	82	通讯对象更新	
	84	原始数据	
	85	参数	
	87	物理地址	
	88	制造商 ID	
	8C	重启	
	8D	编程状态	

8.ETS 参数

通用版数据库的参数的类型和地址如下表。

参数类型	首地址	最大末地址	最大个数
通讯对象使能	3B00	3BF9	250
限定选择类参数	3C00	3CFF	256
1byte 输入类参数	3D00	3DFF	256
2byte 输入类参数	3E00	3EFF	128
4byte 输入类参数	3F00	3FFF	64

例， 限定选择类参数的“功能模式[P3]”地址 = $0x3B00 + 2 = 0x3B02$;

2byte 输入类参数的“2byte 参数 5”地址 = $0x3E00 + 4 = 0x3E04$ 。

注：当前数据库版本只使用了其中一部分空间，即首地址是确定的，实际末地址比表中的小。自己做数据库时，只要不超过上表中的范围即可。

其中，限定选择类参数只能选择固定的几个值，第 1 项值为 0，第 2 项值为 1，如此类推。

9. 通讯对象

通用版数据库共提供了 120 个通讯对象，包括 1bit、4bit、1byte、2byte、4byte，如下表。需要在参数中使能对应的通讯对象才能开启。

类型	通讯对象号范围	数量
1bit	0~31	32
4bit	32~63	32
1byte	100~131	32
2byte	200~215	16
4byte	240~247	8

序号 ^	名称	对象功能	描述	群组地址	长度	C	R	W	T	U	数据类型	优先级
0	1bit Object 1				1 bit	C	R	W	T	-		低
31	1bit Object 32				1 bit	C	R	W	T	-		低
32	4bit Object 1				4 bit	C	R	W	T	-		低
63	4bit Object 32				4 bit	C	R	W	T	-		低
100	1byte Object 1				1 byte	C	R	W	T	-		低
131	1byte Object 32				1 byte	C	R	W	T	-		低
200	2byte Object 1				2 bytes	C	R	W	T	-		低
215	2byte Object 16				2 bytes	C	R	W	T	-		低
240	4byte Object 1	4 bytes			4 bytes	C	R	W	T	-		低
247	4byte Object 8	4 bytes			4 bytes	C	R	W	T	-		低

上图中的序号指的是通讯对象号。

10.ETS 数据库制作

（本段内容仅适于开发数据库者阅读；需要加密狗、下载器可联系）

随板已提供通用版的数据库（.knxproj），此数据库也是可以用的，只是没有针对性。其中的参数和通讯对象的名称、表现形式不人性化，可能需要用另外的文档进行说明自己的定义。如果不使用通用版的数据库，也可以按以下要求开发自己的数据库，同时可支持更多的参数和通讯对象。请使用提供的 S19 文件。

参数地址的范围：0x3B00~0x3FFF，共 1280 字节，自行分配地址。

通讯对象的范围：0~249，共 250 个，数据类型自己定义。

数据库仅用于下载参数的组地址，没有应用程序！

11.拿到板子了不知道怎么开始？

主要工具：ETS 软件、KNX 下载器、KNX 电源、USB 转 TTL 及电脑等。

1)测试 KNX 通讯是否正常

核心板接入 KNX 电源并通过下载器连接电脑，打开 ETS 软件的群组监视器，检查是否有设备处于编程模式。按下核心板的编程按键以点亮编程灯，然后应该可以在 ETS 上看到核心板的设备地址，说明 KNX 通讯正常。

2)测试串口通讯是否正常

用 USB 转 TTL 连接核心板的串口，打开串口助手软件连接（默认波特率 115200，无校验，1 位停止位）。发送重启报文:0C 05 70 70 83 74 6F 91，核心板应该返回开机提示。

3)测试数据库

将测试数据库导入 ETS，选择设备号为 1.1.1 的设备，完整下载。

下载完成后在串口助手上发送：03 03 06 01 01 06，这时在 ETS 上应该能看到通讯对象 6 绑定的组地址发出 1。

在 ETS 上对该组地址写 1,这里在串口助手上应该能看到对应的数据。

4)如何建立自己的 KNX 设备

举个栗子，做一个简单的 8 通道继电器执行器。

其中用到 8 个 1bit 的开关控制通讯对象，收到 1 时吸合继电器，收到 0 时断开继电器，使用 Obj0~Obj7；另外用到 8 个 1bit 的开关状态反馈通讯对象，吸合状态值为 1，断开状态值为 0，使用 Obj8~Obj15。

数据库设置：在 ETS 上开启通讯对象 0~15，为了方便查看可以增加描述，如下下图。

序号 *	名称	对象功能	描述	群组地址	长度	C	R	W	T	U	数据类型	优先级
0	1bit Object 1		通道1开关		1 bit	C	R	W	T	-		低
1	1bit Object 2		通道2开关		1 bit	C	R	W	T	-		低
2	1bit Object 3		通道3开关		1 bit	C	R	W	T	-		低
3	1bit Object 4		通道4开关		1 bit	C	R	W	T	-		低
4	1bit Object 5		通道5开关		1 bit	C	R	W	T	-		低
5	1bit Object 6		通道6开关		1 bit	C	R	W	T	-		低
6	1bit Object 7		通道7开关		1 bit	C	R	W	T	-		低
7	1bit Object 8		通道8开关		1 bit	C	R	W	T	-		低
8	1bit Object 9		通道1反馈		1 bit	C	R	W	T	-		低
9	1bit Object 10		通道2反馈		1 bit	C	R	W	T	-		低
10	1bit Object 11		通道3反馈		1 bit	C	R	W	T	-		低
11	1bit Object 12		通道4反馈		1 bit	C	R	W	T	-		低
12	1bit Object 13		通道5反馈		1 bit	C	R	W	T	-		低
13	1bit Object 14		通道6反馈		1 bit	C	R	W	T	-		低
14	1bit Object 15		通道7反馈		1 bit	C	R	W	T	-		低
15	1bit Object 16		通道8反馈		1 bit	C	R	W	T	-		低

串口命令处理：一、监视通讯对象更新命令(0x82)的 Obj0~Obj7 的数据，收到 1 时控制相应的继电器吸合，收到 0 时控制断开；二、继电器状态变为吸合时通过写通讯对象命令(0x03)向对应的通讯对象 Obj8~Obj15 写 1，状态变为断开时写 0。这样就把这个设备给建立完成了。

使用与测试：将 Obj0~Obj15 分别绑定到 15 个不同的组地址（可以与开关面板关联），然后下载。在 ETS 上对组地址进行读写操作，如果关联了开关面板则可直接操作面板看一下功能是否能实现。

KNX 核心板功能测试板：

