

TreeOS.AI 自动编程系统使用 50 问

1、 TreeOS.AI 是如何实现自动编程的？

TreeOS.AI 的自动编程功能是基于对原理图中各器件名称及它们之间网络连接的识别，以及 MCU 和各种外围器件的构件化代码库。可自动生成 MCU 器件引脚配置代码、外围器件初始化等驱动代码。生成的代码集成 TreeOS 操作系统，可支持高、中、低速全系列 MCU。

2、 TreeOS.AI 对上传的电路图（网络表文件）有何要求？

对于普通功能芯片（或模块）来说，在绘制原理图时，应保证器件的注释与芯片（或模块）的名称一致，各引脚标号与芯片（或模块）厂商发布的数据手册（说明书）一致，方可正常识别。但对于某些器件而言，需要标注特定内容，通过标识符与注释相结合的方法，识别器件及其特定运行参数，实现自动编程。

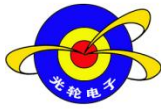
3、 原理图中器件的连接网络命名有何要求？

TreeOS.AI 对器件功能引脚的连接网络命名无任何特别要求。

4、 原理图中的电源、地等网络命名有何要求？

系统可识别的合法电源网络名包括(忽略大小写):“VCC”,“+5V”,“+3V3”,“+3.3V”,“+3V”,“5V”,“3V”,“3V3”,以“VCC”结尾的网络名;

系统可识别的合法“地”网络名（忽略大小写）:“GND”,“VSS”,以“GND”开始或结尾的网络。



5、 TreeOS.AI 是否支持多个原理图文件构成的原理图工程？

TreeOS.AI 完全支持多个原理图文件构成的原理图工程。此时，就需要在 AD 软件中导出整个工程的网络表。点击 Design->Netlist For Project->PCADnlt，就可导出所需的工程网络表文件。

6、 原理图中 MCU 标识符和注释有何要求？

原理图中 MCU 标识符命名无要求，但其注释需符合一定规范格式，即：MCU 具体型号+”_主时钟频率 MHz”，例如：STM32F103RBT_72MHz。MCU 具体型号必须精确到与器件引脚排列相关的尾缀，例如：STM32F103 系列 MCU，103 之后的“C”、“8”、“T”与芯片的封装和引脚排列相关，不可省略，这三位字符之后的内容则可以省略，不影响识别。”_主时钟频率”是指 MCU 的运行主时钟频率。如果”_主时钟频率”项目空缺，则按照默认时钟生成代码。

7、 如果原理图中有多个 MCU 可以分别对它们自动编程吗？

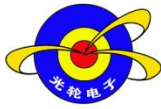
目前 TreeOS.AI 智能对单个 MCU 自动编程，未来可实现对原理图中的多个 MCU 识别和自动编程。

8、 原理图中电阻的标识符和注释有何要求？

电阻的标识符命名需要以” R”或”r”开头，注释的合法字符包括：“0-9”，“R”，“.”，”K”，“k”，“M”，“m”等。

9、 TreeOS.AI 能否正常识别电阻上拉、下拉的高低电平？

完全可以，但电阻的标识符和注释一定符合要求。



10、 TreeOS.AI 能否识别电路中的跳线？

需要将跳线设计为 0 欧姆电阻，系统方能识别。

11、 如果信号线中串入电阻能否正常识别？

完全可以，但电阻的标识符和注释一定符合要求。

12、 原理图中电容的标识符和注释有何要求？

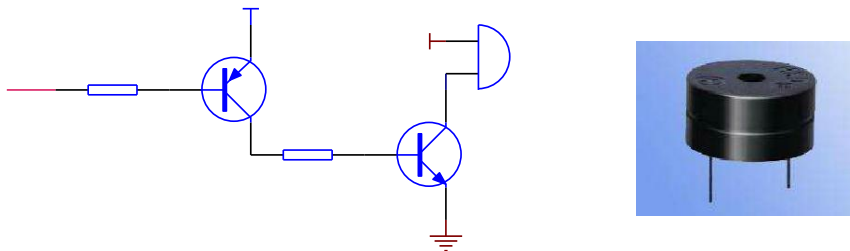
电容的标识符命名需要以“C”或“c”开头，注释的合法字符包括：

“0-9”，“F”，“.”，”u”，“n”，“p”等。

13、 三极管是否可以识别，其命名和连接方式有何要求？

完全可以，目前可识别的三极管型号有 8050, 9013, 9014, 8550, 9012 等，其注释命名应与型号同名。

目前，系统可识别用于构成驱动电路的三极管，如下图：

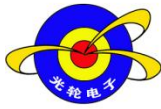


14、 运算放大器芯片能否识别？

可以，目前支持的运算放大器芯片为 LM358。今后，还会支持更多芯片。

15、 MCU 的片上 ADC 功能代码是否可以自动生成？

可以，当 MCU 的片上 ADC 引脚连接到电容或运算放大电路的



输出，则对应引脚被识别为 ADC 功能，进而生成对应的功能代码。

16、MCU 的片上 PWM 功能代码是否可以自动生成？

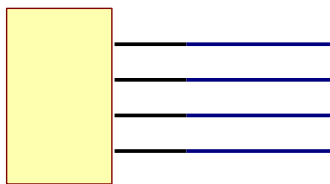
可以，目前当 MCU 的片上 PWM 引脚连接到高速光耦芯片 6N137 时，则自动识别为 PWM 功能。

17、MCU 的片上 UART 串口功能是否可以识别，并实现自动编程？

可以，当 MCU 的 UART 串口连接到相应的器件或模块时，可自动生成对应的功能代码，并根据器件或模块自动设置所需波特率。

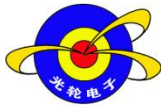
18、如果将 MCU 上的 UART 串口通过接插件直接引出作调试用，能否自动生成相关代码？

可以，需要将 UART 串口连接的接线端子等接口器件标识符命名为 MCU_UART，如下图所示



需要强调的是，这里对接线端子各个引脚的编号等无任何限制。

19、I2C，SPI 等标准接口器件是否一定要连接到 MCU 对应的通信功能接口？



不需要，为了提高可靠性和通用性，I2C，SPI 等标准接口器件均采用模拟总线时序控制模式，并为采用 MCU 的片上 I2C，SPI 通信控制器。

20、 所有 I2C 总线器件一定要挂在同一对信号线上吗？

不需要，代码中每个器件的 I2C 总线都会单独命名。

21、 LED 指示灯是否可以识别，并实现自动编程？

可以，LED 指示灯的标识符命名需要设置为“LED”，连接到 LED 指示灯的 MCUIO 口会配置为输出，并在 TreeOS_Digital_Out.h 中定义 LED_ON 和 LED_OFF，供使用者在应用软件中调用，实现 LED 指示灯的开关。

22、 数码管是否可以识别，并实现自动编程？

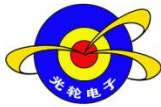
目前，系统可识别的数码管为 4 位七段式扫描数码管（共阴），型号是 SM420364。

23、 目前系统只可实现共阴数码管驱动代码的自动编程，如采用共阳数码管可否实现自动编程？

目前，系统可识别的数码管为 4 位七段式扫描数码管（共阴），型号是 SM420364。如采用共阳数码管，其型号也应在图中设置为 SM420364，并需要手动修改相关代码。

24、 继电器可否识别，并实现编程？

可以，目前可识别的继电器型号为 JZC-32F/012-ZS3(555)，目前



系统只能识别与单片机连接的继电器线圈，触点信号无法识别。

25、 如果采用的继电器不在支持器件的列表中怎么办？

可以将器件在原理图中命名为 Relay。

26、 有源蜂鸣器可否识别，并实现自动编程？

可以，其原理图注释需要命名为 BEEP_DC。

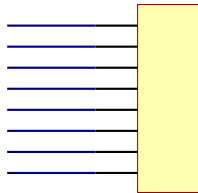
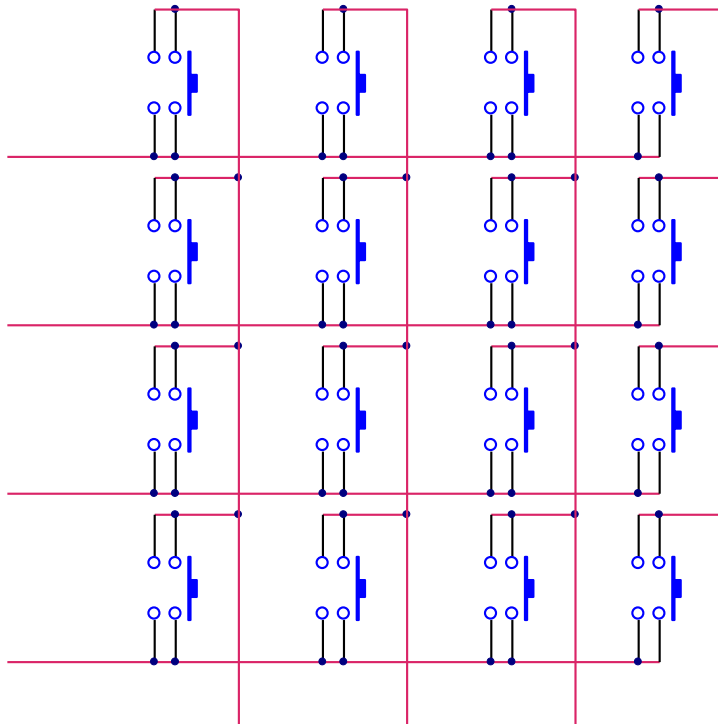
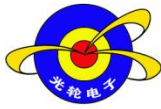
27、 无源蜂鸣器可否识别，并实现自动编程？

目前还不可以，预计将在以后的版本中实现。

28、 矩阵键盘可否识别，并实现自动编程？

可以实现 4 乘 4 或 3 乘 3 矩阵键盘的识别和自动编程，但需要注意，整个键盘作为一个器件命名必须为 Keyboard4x4 或 Keyboard3x3。

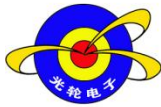
如通过单个按键组合实现的矩阵键盘，则需要额外添加一个键盘器件，如下图所示(P2 为额外添加用于 TreeOS.AI 自动编程的器件)：



其引脚顺序需要满足一定要求。原理图中的4x4键盘为8个引脚，引脚1至4号依次为第1列至第4列（从左至右）的公共端，第5至8号依次为第1行至第4行（从上至下）的公共端；3x3键盘为6个引脚，引脚1至3号依次为第1列至第3列（从左至右）的公共端，第4至6号依次为第1行至第3行（从上至下）的公共端。

29、单按键可否识别，并实现自动编程？

可以。按键的标识符名需设定为 **Key**，其一端接地，另一端接单



片机 IO 口即可。

30、 缓冲器或多路锁存器可否识别，并实现自动编程？

可以，目前可识别的缓冲器或多路锁存器有：74HC574 和 74HC573。

31、 缓冲器或锁存器的 OE 信号是否必须连接 MCU？

不是必须的，自动编程系统会根据连接情况自动生成代码。

32、 总线型缓冲器或锁存器所有接口引脚是否都要用到？

不是必须的，但是要保证最低 bit 和最高 bit 有 1 位使用。例如：74HC574 有 D0~D7 共 8 个锁存器，D0 和 D7 需要至少用 1 个。

33、 目前支持的单色液晶显示屏有哪些？

目前支持的单色液晶有：

器件型号或名称	原理图中应采用的注释	功能
122×32 液晶	LCD_12232_SED1520	并口，主控芯片为 SED1520
128×64 液晶	LCD_12864_HD61202	并口，主控芯片为 HD61202
16 字符×2 行液晶	LCD_1602	并口，主控芯片为 HD44780

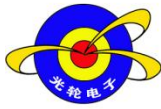
34、 TreeOS.AI 是否支持信号串并转换移位寄存器？

目前 TreeOS.AI 支持的信号串并转换移位寄存器为 74HC595。

35、 TTL 转 RS232 等电平转换电路可否识别，并实现自动编程？

完全可以。

36、 如果 MCU 的 RAM 或 FLASH 等内部资源不足或运算速度跟不上，能否正常实现自动编程？



这属于硬件设计和选型问题，TreeOS.AI 可照常生成代码，但不能保证其在此情况下能正常可靠运行。

37、 TreeOS.AI 能否识别一个系统中的多个相同型号器件？

部分器件支持。在文档《TreeOS.AI 使用说明书》的支持器件表格中，如果支持多器件列为是，则支持多个相同信号器件，如果为否，则表示不支持。

为了实现多个相同信号器件的自动编程，系统运行时为每个芯片或模块分配一个 TreeOS 编码，编码为 0, 1, 2, 3...对应第 1, 2, 3, 4...个器件。在调用相关功能函数时需要传入对应参数。

38、 LED 和数码管等器件通过锁存器、缓冲器等器件连接到 MCU 能否正常识别并生成代码？

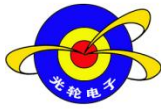
完全可以，这对应用软件完全透明，此时只需调用 LED 等器件的控制函数即可。

39、 传感器芯片和其相对应的模块识别有何区别？

传感器模块就是将传感器芯片安装在一块电路板上，只在模块引出必须的信号线，其引脚顺序自然会与芯片有所不同，使用时应注意。

40、 TreeOS.AI 生成的代码能否直接用于项目开发？

完全可以，TreeOS.AI 代码完全开源，但所提供的所有程序仅供购买者本人作为学习之用，其目的是为了您能更好地学习和理解 TreeOS 实时操作系统。若您自行把本软件引用到您的产品之中，运



行后出现由本软件直接或间接引起的任何不良后果，本软件著作权所有者将不会作出任何赔偿，亦不承担任何法律责任。

41、 TreeOS.AI 生成的代码能否用于 $\mu\text{C}/\text{OS}$ 等实时操作系统？

完全可以，TreeOS.AI 代码完全开源，但所提供的所有程序仅供购买者本人作为学习之用，其目的是为了您能更好地学习和理解 TreeOS 实时操作系统。若您自行把本软件引用到您的产品之中，运行后出现由本软件直接或间接引起的任何不良后果，本软件著作权所有者将不会作出任何赔偿，亦不承担任何法律责任。

42、 TreeOS.AI 支持哪些嵌入式开发软件？

鉴于大多数开发者习惯于在 Keil 上做开发，本系统目前只支持 Keil 开发平台。

43、 MCU 的外部中断代码是否可以自动生成？

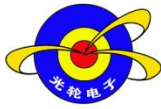
用到了外部中断的器件，例如超声测距模块（AJ-SR04M 模块）等，均可通过本系统自动生成相应外部中断代码，但是需要注意的是硬件设计时，器件相应的引脚必须连接到 MCU 的外部中断引脚。

44、 TreeOS.AI 是否支持红外遥控信号接收的自动编程？

支持。目前支持的红外遥控信号接收头有：IRM3638T 和 HS0038。

45、 必须采用指定的红外遥控信号接收头才能实现自动编程？

如果采用的红外遥控接收头不在支持的器件列表中，需要在生成代码时，暂时将原理图中的器件改为列表中的器件名。



46、 对并行数据总线与 MCU 的 IO 口连接有何要求？

无任何要求，完全可以根据需要为总线分配空闲的 IO 口即可，但如果利用属于多组总线的 IO 口凑成一个 8 位并行数据总线，可能会降低 MCU 执行速度。

47、 是否支持多个器件共享并行数据总线？

完全支持，且建议锁存器，模块化并口 LCD 显示器共用一组并行数据总线。

48、 TreeOS.AI 是否支持 51 单片机的外挂 RAM 或 ROM？

鉴于目前 MCU 片上存储器资源已可以满足大部分的应用需求，目前暂不支持所有单片机的外挂 RAM 或 ROM。

49、 电源模块等与单片机无关的电路是否会影响 TreeOS.AI 运行？

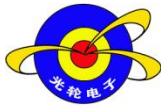
完全不会影响。

50、 是否支持 MCU 的内部时钟振荡器？如何设置 MCU 的运行主频？

支持，但需要在 MCU 的注释中设置其运行主频。此外，MCU 内部时钟振荡器的设置一般需要通过熔丝配置等非软件的方式实现，本系统无法操作。

STM32F103 单片机的运行主频默认为 72MHz，无法修改，且必须外挂数据手册建议的 8MHz 晶振运行。

51、 TreeOS.AI 能否自动识别晶振器件？



不能。因此需要在 MCU 的注释中设置其运行主频。

52、 生成的代码是否提供说明？

在生成代码压缩包的根目录下，提供了一个 Readme.txt 文件，包括了网络表文件信息和生成的代码模块使用说明，具体内容如下：

```
Readme - Notepad
-----
File Edit Format View Help
欢迎使用TreeOS.AI代码生成系统 版本V2.0 2019-07-11

1.代码生成信息:
原理图网络文件: MCU For STC12C5A60S2.NET
上传时间: 2019-08-01 15:01:53
MCU型号: STC12C5A60S2_35i_LQPF44G 标识符: U1

器件/MCU片上外设TreeOS编号:
标识符 = U1; 器件型号 = TM7705; TreeOS编号 = 0;
标识符 = BEEP; 器件型号 = BEEP_DC; TreeOS编号 = 0;

2. TreeOS库函数使用说明:

TreeOS_tm7705.c/.h
此文件为TM7705的驱动代码，外部函数及使用方法:
(1).void TM7705_Init(void)
上电初始化函数，初始化TM7705总线接口，同步SPI接口时钟时序，进行自校准一次，在TreeOS_main.c的最开始部分调用（已经在代码中自动生成）

(2).void TM7705_WriteReg(ui8_RegID, uint32_t_RegValue)
写寄存器函数，_RegID:寄存器ID, _RegValue:寄存器值对于8位的寄存器，取32位形参的低8bit

(3).uint32_t TM7705_ReadReg(ui8_RegID)
读寄存器函数，_RegID: 寄存器ID

(4).void TM7705_CalibSelf(ui8 _ch)
自校验函数，_ch: ADC通道, 1或2

(5).void TM7705_SytemCalibZero(ui8 _ch)
零位校准函数，_ch: ADC通道, 1或2

(6).void TM7705_SytemCalibFull(ui8 _ch)
满位校准函数，_ch: ADC通道, 1或2

(7).ui16 TM7705_ReadAdc(ui8 _ch)
读取TM7705ADC函数，输入参数_ch: ADC通道, 1或2，返回读到的16位ADC数据

TreeOS_BEEP.c/.h
此文件为BEEP的中断驱动代码，外部函数及使用方法:
(1).void set_beep(ui8 btype)
BEEP鸣叫函数，输入参数为鸣叫类型，有关闭（NOT_BEEP）、短鸣（SHORT_BEEP）、长鸣（LONG_BEEP）、间隔鸣（SPACING_BEEP）四种类型，鸣叫参数可以在h文件中修改

(2).void scan_beep(void)
定时检查有源蜂鸣器的状态是否需要改变函数，在scan_in_while()调用（已经在代码中自动生成）

(3).void beep_init(void)
上电初始化函数，在TreeOS_main.c的最开始部分调用（已经在代码中自动生成）
```

A.欢迎信息

B.代码生成信息

C.生成的库函数使用说明