

## 前言

自举程序存储在 PY32 器件的内部自举 ROM 存储器 (系统存储器) 中。在生产期间由 Puya 编程。其主要任务是通过一种可用的串行外设 (USART、CAN、USB、I2C 等) 将应用程序下载到内部 Flash 中。每种串行接口都定义了相应的通信协议，其中包含兼容的命令集和序列。

本文档中所列产品在整个文档中称为 PY32。

自举程序的主要功能如下：

- 使用嵌入式串行接口按照预定义的通信协议下载代码
- 可传送并更新 Flash 代码、数据和向量表部分

本应用笔记介绍了自举程序的一般概念。说明了使用 PY32 器件的自举程序时支持的外设以及需要考虑的硬件要求。不过，支持的每种串行外设的低层级通信协议规范在单独的文档中进行了介绍。

## 目录

<b>1</b>	<b>相关文档</b>	<b>3</b>
<b>2</b>	<b>自举程序一般说明</b>	<b>4</b>
2.1	自举程序激活	4
2.2	退出系统存储器自举模式	4
2.3	自举程序标识	4
<b>3</b>	<b>器件相关的自举程序参数</b>	<b>6</b>
<b>4</b>	<b>PY32F002A、PY32F003、PY32F030、PY32M030 器件自举程序</b>	<b>7</b>
4.1	自举程序配置	7
4.2	自举程序硬件要求	7
4.3	自举程序选择	7
4.4	自举程序版本	8
<b>5</b>	<b>PY32F031 器件自举程序</b>	<b>9</b>
5.1	自举程序配置	9
5.2	自举程序硬件要求	9
5.3	自举程序选择	9
5.4	自举程序版本	10
<b>6</b>	<b>PY32F040 器件自举程序</b>	<b>11</b>
6.1	自举程序配置	11
6.2	自举程序硬件要求	12
6.3	自举程序选择	12
6.4	自举程序版本	12
<b>7</b>	<b>PY32F07x、PY32M07x 器件自举程序</b>	<b>13</b>
7.1	自举程序配置	13
7.2	自举程序硬件要求	14
7.3	自举程序选择	14
7.4	自举程序版本	14
<b>8</b>	<b>PY32F403 器件自举程序</b>	<b>15</b>
8.1	自举程序配置	15
8.2	自举程序硬件要求	16
8.3	自举程序选择	16
8.4	自举程序版本	16
<b>9</b>	<b>版本历史</b>	<b>17</b>

## 1 相关文档

有关所支持的每个产品（如表 1 中所列）的信息，请参见 <https://www.puyasemi.com/> 提供的以下文档：

- 数据手册或产品简介
- 参考手册和/或 *flash* 编程手册

PUYA CONFIDENTIAL

2 自举程序一般说明

2.1 自举程序激活

通过配置特定“系统存储器”配置中的 BOOT0 和 BOOT1 引脚（请参见表 2.1-1），然后执行复位即可自动激活自举程序。

根据所用引脚配置的不同，可选择 Flash、系统存储器或 SRAM 作为自举空间，如以下表 2.1-1 所示。在一些产品中，BOOT1 不表示 I/O 引脚，而是选项字节区域的一个位。PY32F0xx 和 PY32M0x 器件就属于这种情况，此时 BOOT1 通过选项字节中的 nBoot1 位进行配置。

- 当 nBoot1 位置 1 时，相当于表 2.1-1 中的 BOOT1 复位为 0。
- 当 nBoot1 位复位为 0 时，相当于表 2.1-1 中的 BOOT1 置 1。

表 2.1-1.自举引脚配置

自举模式选择引脚		自举模式	别名使用
BOOT1	BOOT0		
X	0	用户 Flash	选择用户 Flash 作为自举空间
0	1	系统存储器	选择系统存储器作为自举空间
1	1	嵌入式 SRAM	选择嵌入式 SRAM 作为自举空间

表 2.1-1 表明 BOOT 引脚配置如下时，PY32 微控制器进入系统存储器自举模式：

- BOOT0 = 1
- BOOT1 = 0

复位后，在 SYSClk 的第四个上升沿锁存 BOOT 引脚的值。

注： 在一些产品中，当 PY32 产品提供双存储区自举特性功能时，(BOOT0 = 0 且 BOOT1 = x) 可进入自举程序。有关详细信息，请参见产品部分的双存储区自举特性部分。

当工作温度超出环境温度范围时，由于内部时钟 (HSI) 随着温度变化，致使串行通信协议时钟损坏，因而可能导致自举程序无法正常运行。

2.2 退出系统存储器自举模式

为了执行应用程序，必须先退出系统存储器自举模式。执行硬件复位即可实现此操作。复位期间，BOOT 引脚/位 (BOOT0 和 BOOT1) 必须设置为合适的电平，以选择所需的自举模式 (请参见表 2.1-1)。复位后，CPU 将从自举存储器的存储器地址空间最底部（起始地址 0x0000 0000）开始执行代码。

2.3 自举程序标识

根据所用 PY32 器件的不同，自举程序可支持一个或多个嵌入式串行外设，用来将代码下载到内部 Flash 中。自举程序标识符 (ID) 可提供有关支持的串行外设的信息。

对于给定的 PY32 器件，自举程序通过以下各项标识：

1. **自举程序（协议）版本：**自举程序中使用的串行外设（USART、CAN、USB 等）通信协议的版本。可使用自举程序 Get Version 命令检索此版本。
2. **自举程序标识符(ID)：**PY32 器件自举程序的版本，以 0xXY 形式的单字节代码表示，其中：
  - X 指定器件自举程序所用的嵌入式串行外设：
    - X = 1: 使用一个 USART
    - X = 2: 使用两个 USART
    - X = 3: 使用 USART、CAN 和 DFU
    - X = 4: 使用 USART 和 DFU
    - X = 5: 使用 USART 和 I2C

- X = 6: 使用 I2C  
 X = 7: 使用 USART、CAN、DFU 和 I2C  
 X = 8: 使用 I2C 和 SPI  
 X = 9: 使用 USART、CAN (或 FDCAN)、DFU、I2C 和 SPI  
 X = 10: 使用 USART、DFU 和 I2C  
 X = 11: 使用 USART、I2C 和 SPI  
 X = 12: 使用 USART 和 SPI  
 X = 13: 使用 USART、DFU、I2C 和 SPI

- Y 指定器件的自举程序版本

下面以自举程序 ID 0x10 为例。这表示仅使用一个 USART 的器件自举程序的第一个版本。  
 自举程序 ID 编程在器件系统存储器最后一个字节地址减 1 所对应的空间中, 可通过自举程序  
 “Read memory”命令来读取, 或者通过使用 JTAG/SWD 直接访问系统存储器来读取。

下表提供了有关 PY32 器件嵌入式自举程序的标识信息。

表 2.3-1. 嵌入式自举程序

PY32 系列	器件	支持的串行外设	自举程序 ID		自举程序 (协议) 版本
			ID	存储单元	
F0	PY32F002Axx/ PY32F003xx/ PY32F030xx	USART1	V1.0	0x1FFF0000	USART(V1.0)
	PY32F040xx/ PY32F071xx/ PY32F072xx	USART1/USART2/ USART3/USART4/ I2C1/DFU(USB 设备 FS)	V1.0	0x1FFF0000	USART(V1.0)/ I2C(V1.0)/ DFU(V1.0)
F4	PY32F403xx	USART1/USART2/ USART3/USART4/ I2C1/DFU(USB 设备 FS)	V1.0	0x1FFF0000	USART(V1.0)/ I2C(V1.0)/ DFU(V1.0)
M0	PY32M030xx	USART1	V1.0	0x1FFF0000	USART(V1.0)
	PY32M070xx	USART1/USART2/ USART3/USART4/ I2C1/DFU(USB 设备 FS)	V1.0	0x1FFF0000	USART(V1.0)/ I2C(V1.0)/ DFU(V1.0)

### 3 器件相关的自举程序参数

对于所有 PY32 器件，每种串行外设（USART、CAN、USB 和 I2C）的自举程序协议命令集和序列都相同。但某些参数与器件相关。对一些命令，某些参数值可能取决于所使用的器件。这些参数如下：

- PID（产品 ID），该参数因器件而异
- 支持 Read Memory、Go 和 Write Memory 命令时，自举程序支持的有效存储器地址（RAM、Flash、系统存储器、选项字节区域）。
- 执行 Write Protect 命令时使用的 Flash 扇区的大小。

下表显示了生产中的每种 PY32 器件自举程序的上述参数的值。

表 3.3-1. 器件相关的自举程序参数

PY32 系列	器件	产品 (器件) ID	SRAM 存储器	Flash	Flash 扇区大小	选项字节区域	系统存储器
F0	PY32F002Axx	0x0440	0x20000200-0x20000BFF	0x08000000-0x08004FFF	4KB(32 页, 每页 128B)	0x1FFF0E80-0x1FFF0E8F	0x1FFF0000-0x1FFF0D7F
	PY32F003xx	0x0440	0x20000200-0x20001FFF	0x08000000-0x0800FFFF	4KB(32 页, 每页 128B)	0x1FFF0E80-0x1FFF0E8F	0x1FFF0000-0x1FFF0D7F
	PY32F030xx	0x0440	0x20000200-0x20001FFF	0x08000000-0x0800FFFF	4KB(32 页, 每页 128B)	0x1FFF0E80-0x1FFF0E8F	0x1FFF0000-0x1FFF0D7F
	PY32F040xx	0x0448	0x20000800-0x20003FFF	0x08000000-0x0801FFFF	8KB(32 页, 每页 256B)	0x1FFF3100-0x1FFF311F	0x1FFF0000-0x1FFF2EFF
	PY32F071xx	0x0448	0x20000800-0x20003FFF	0x08000000-0x0801FFFF	8KB(32 页, 每页 256B)	0x1FFF3100-0x1FFF311F	0x1FFF0000-0x1FFF2EFF
	PY32F072xx	0x0448	0x20000800-0x20003FFF	0x08000000-0x0801FFFF	8KB(32 页, 每页 256B)	0x1FFF3100-0x1FFF311F	0x1FFF0000-0x1FFF2EFF
F4	PY32F403xx	0x0413	0x20000800-0x2000FFFF	0x08000000-0x0805FFFF	2KB(8 页, 每页 256B)	0x1FFF5000-0x1FFF500F	0x1FFF0000-0x1FFF4FFF
M0	PY32M030xx	0x0440	0x20000200-0x20001FFF	0x08000000-0x0800FFFF	4KB(32 页, 每页 128B)	0x1FFF0E80-0x1FFF0E8F	0x1FFF0000-0x1FFF0D7F
	PY32M070xx	0x0448	0x20000800-0x20003FFF	0x08000000-0x0801FFFF	8KB(32 页, 每页 256B)	0x1FFF3100-0x1FFF311F	0x1FFF0000-0x1FFF2EFF

## 4 PY32F002A、PY32F003、PY32F030、PY32M030 器件自举程序

### 4.1 自举程序配置

PY32F002A、PY32F003、PY32F030、PY32M030 器件嵌入式自举程序仅支持 USART1 这一种接口。下表介绍了系统存储器自举模式下自举程序需要使用的硬件资源。

表 4.1-1.系统存储器自举模式下 PY32 器件的配置

自举程序	功能/外设	状态	备注
USART1 自举程序	时钟源	HSI 使能	使用 HSI 作为系统时钟，频率为 24MHz
	RAM	-	自地址 0x20000000 起的 512 个字节空间供自举程序使用。
	系统存储器	-	自地址 0x1FFF0000 起的 3.25KB 空间包含自举程序固件。
	IWDG	-	独立看门狗(IWDG)预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件 IWDG 选项）。
	USART1	使能	初始化后，USART1 的配置为:8 位，偶校验位和 1 个停止位。
	USART1 自动波特率检测	使能	用于自动检测主机串口波特率。
第 1 组 USART1	USART1_RX 引脚	输入	PA3 引脚: USART1 用于接收。
	USART1_TX 引脚	推挽输出	PA2 引脚: USART1 用于发送。
	检测阶段期间，USART1_RX(PA10)、USART1_RX(PA15)引脚必须保持高电平或低电平。		
第 2 组 USART1	USART1_RX 引脚	输入	PA10 引脚: USART1 用于接收。
	USART1_TX 引脚	推挽输出	PA9 引脚: USART1 用于发送。
	检测阶段期间，USART1_RX(PA3)、USART1_RX(PA15)引脚必须保持高电平或低电平。		
第 3 组 USART1	USART1_RX 引脚	输入	PA15 引脚: USART1 用于接收。
	USART1_TX 引脚	推挽输出	PA14 引脚: USART1 用于发送。
	检测阶段期间，USART1_RX(PA3)、USART1_RX(PA10)引脚必须保持高电平或低电平。		

系统时钟由嵌入式内部高速 RC 提供，自举程序代码不需要外部石英时钟。

下载应用程序二进制代码后，如果选择执行 Go 命令，则在跳转到用户应用程序之前，自举程序使用的外设寄存器（如上表所示）不会初始化为默认复位值。如果要使用这些寄存器，应在用户应用程序中对其重新配置。因此，如果应用程序正在使用 IWDG，则必须调整 IWDG 预分频器值来满足应用程序的要求（因为自举程序已将预分频器设置为最大值）。

### 4.2 自举程序硬件要求

将 PY32 置于系统存储器自举模式所需的硬件包含任何相关电路、开关或跳线，它们能够在复位期间使 BOOT0 引脚保持高电平，使 BOOT1 引脚保持低电平。

要在系统存储器自举模式下连接 PY32，必须将 RS232 串行接口直接连接到 USART1\_RX (PA10) 和 USART1\_TX (PA9) 引脚。

注：USART1\_CK、USART1\_CTS 和 USART1\_RTS 引脚未使用，因此用户可将这些引脚用于其它外设或 GPIO。

### 4.3 自举程序选择

进入系统存储器自举模式且微控制器已按上述要求配置后，自举程序代码立即开始扫描 USART1\_RX 引脚，等待接收 0x7F 数据帧：一个起始位、0x7F 数据位、偶校验位和一个停止位。

此数据帧的起始位用于判断使用哪一组 USART1。此数据帧的数据位则用于串口的自动波特率检测。

随后，代码将相应初始化串行接口。通过计算出的波特率，向主机返回确认字节 (0x79)，指示 PY32 已准备好接收用户命令。

4.4 自举程序版本

下表列出了 PY32F002A、PY32F003、PY32F030、PY32M030 器件的自举程序版本。

表 4.4-1. 自举程序版本

自举程序版本号	说明
V1.0	初始自举程序版本。



5 PY32F031 器件自举程序

5.1 自举程序配置

PY32F031 器件嵌入式自举程序仅支持 USART1 这一种接口。  
下表介绍了系统存储器自举模式下自举程序需要使用的硬件资源。

表 5.1-1.系统存储器自举模式下 PY32 器件的配置

自举程序	功能/外设	状态	备注
USART1 自举程序	时钟源	HSI 使能	使用 HSI 作为系统时钟，频率为 24MHz
	RAM	-	自地址 0x20000000 起的 512 个字节空间供自举程序使用。
	系统存储器	-	自地址 0x1FFF0000 起的 3.25KB 空间包含自举程序固件。
	IWDG	-	独立看门狗(IWDG)预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件 IWDG 选项）。
	USART1	使能	初始化后，USART1 的配置为:8 位，偶校验位和 1 个停止位。
	USART1 自动波特率检测	使能	用于自动检测主机串口波特率。
第 1 组 USART1	USART1_RX 引脚	输入	PA3 引脚：USART1 用于接收。
	USART1_TX 引脚	推挽输出	PA2 引脚：USART1 用于发送。
	检测阶段期间，USART1_RX(PA10)、USART1_RX(PA15)引脚必须保持高电平或低电平。		
第 2 组 USART1	USART1_RX 引脚	输入	PA10 引脚：USART1 用于接收。
	USART1_TX 引脚	推挽输出	PA9 引脚：USART1 用于发送。
	检测阶段期间，USART1_RX(PA3)、USART1_RX(PA15)引脚必须保持高电平或低电平。		
第 3 组 USART1	USART1_RX 引脚	输入	PA15 引脚：USART1 用于接收。
	USART1_TX 引脚	推挽输出	PA14 引脚：USART1 用于发送。
	检测阶段期间，USART1_RX(PA3)、USART1_RX(PA10)引脚必须保持高电平或低电平。		

系统时钟由嵌入式内部高速 RC 提供，自举程序代码不需要外部石英时钟。  
下载应用程序二进制代码后，如果选择执行 Go 命令，则在跳转到用户应用程序之前，自举程序使用的外设寄存器（如上表所示）不会初始化为默认复位值。如果要使用这些寄存器，应在用户应用程序中对其重新配置。因此，如果应用程序正在使用 IWDG，则必须调整 IWDG 预分频器值来满足应用程序的要求（因为自举程序已将预分频器设置为最大值）。

5.2 自举程序硬件要求

将 PY32 置于系统存储器自举模式所需的硬件包含任何相关电路、开关或跳线，它们能够在复位期间使 BOOT0 引脚保持高电平，使 BOOT1 引脚保持低电平。  
要在系统存储器自举模式下连接 PY32，必须将 RS232 串行接口直接连接到 USART1\_RX (PA10) 和 USART1\_TX (PA9) 引脚。  
注：USART1\_CK、USART1\_CTS 和 USART1\_RTS 引脚未使用，因此用户可将这些引脚用于其它外设或 GPIO。

5.3 自举程序选择

进入系统存储器自举模式且微控制器已按上述要求配置后，自举程序代码立即开始扫描 USART1\_RX 引脚，等待接收 0x7F 数据帧：一个起始位、0x7F 数据位、偶校验位和一个停止位。  
此数据帧的起始位用于判断使用哪一组 USART1。此数据帧的数据位则用于串口的自动波特率检测。  
随后，代码将相应初始化串行接口。通过计算出的波特率，向主机返回确认字节 (0x79)，指示 PY32 已准备好接收用户命令。

5.4 自举程序版本

下表列出了 PY32F031 器件的自举程序版本。

表 5.4-1. 自举程序版本

自举程序版本号	说明
V1.0	初始自举程序版本。

## 6 PY32F040 器件自举程序

### 6.1 自举程序配置

PY32F040 器件嵌入式自举程序支持三种串行外设：USART1、USART2、USART3、USART4、I2C。

下表介绍了系统存储器自举模式下 PY32F040 器件的自举程序需要使用的硬件资源。

表 6.1-1.系统存储器自举模式下 PY32 器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	时钟源	HSI 使能	使用 PLL 作为系统时钟，频率为 48MHz
	RAM	-	自地址 0x20000000 起的 2048 个字节空间供自举程序使用。
	系统存储器	-	自地址 0x1FFF0000 起的 12KB 空间包含自举程序固件。
	IWDG	-	独立看门狗(IWDG)预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件 IWDG 选项）。
USART1 自举程序	USART1	使能	初始化后，USART1 的配置为:8 位，偶校验位和 1 个停止位。
	USART1_RX 引脚	输入	PA10 引脚：USART1 用于接收。
	USART1_TX 引脚	推挽输出	PA9 引脚：USART1 用于发送。
	USART1 自动波率检测	使能	用于自动检测主机串口波特率。
	检测阶段期间，USART2_RX(PA15)、USART3_RX(PB11)、USART4_RX(PC11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
USART2 自举程序	USART2	使能	初始化后，USART2 的配置为:8 位，偶校验位和 1 个停止位。
	USART2_RX 引脚	输入	PA15 引脚：USART2 用于接收。
	USART2_TX 引脚	推挽输出	PA14 引脚：USART2 用于发送。
	USART2 自动波率检测	使能	用于自动检测主机串口波特率。
	检测阶段期间，USART1_RX(PA10)、USART3_RX(PB11)、USART4_RX(PC11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
USART3 自举程序	USART3	使能	初始化后，USART3 的配置为:8 位，偶校验位和 1 个停止位。
	USART3_RX 引脚	输入	PB11 引脚：USART3 用于接收。
	USART3_TX 引脚	推挽输出	PB10 引脚：USART3 用于发送。
	USART3 自动波率检测	使能	用于自动检测主机串口波特率。
	检测阶段期间，USART1_RX(PA10)、USART2_RX(PA15)、USART4_RX(PC11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
USART4 自举程序	USART4	使能	初始化后，USART4 的配置为:8 位，偶校验位和 1 个停止位。
	USART4_RX 引脚	输入	PC11 引脚：USART4 用于接收。
	USART4_TX 引脚	推挽输出	PC10 引脚：USART4 用于发送。
	USART4 自动波率检测	使能	用于自动检测主机串口波特率。
	检测阶段期间，USART1_RX(PA10)、USART2_RX(PA15)、USART3_RX(PB11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
I2C1 自举程序	I2C1	使能	PLL 作为 I2C 时钟源。初始化后，I2C1 的配置为：7 位地址，从模式，从地址：0x76。
	I2C1_SCL 引脚	输入/输出	PB6 引脚：I2C1 时钟线在开漏模式下使用。
	I2C1_SDA 引脚	输入/输出	PB7 引脚：I2C1 时钟线在开漏模式下使用。
	检测阶段期间，USART1_RX(PA10)、USART2_RX(PA15)、USART3_RX(PB11)、USART4_RX(PC11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		

系统时钟由嵌入式内部高速 RC 提供，自举程序代码不需要外部石英时钟。

下载应用程序二进制代码后，如果选择执行 Go 命令，则在跳转到用户应用程序之前，自举程序使用的外设寄存器（如上表所示）不会初始化为默认复位值。如果要使用这些寄存器，应在用户应用程序中对其重新配置。因此，如果应用程序正在使用 IWDG，则必须调整 IWDG 预分频器值来满足应用程序的要求（因为自举程序已将预分频器设置为最大值）。

6.2 自举程序硬件要求

将 PY32 置于系统存储器自举模式所需的硬件包含任何相关电路、开关或跳线，它们能够在复位期间使 BOOT0 引脚保持高电平，使 BOOT1 引脚保持低电平。

要在系统存储器自举模式下连接 PY32，必须将 RS232 串行接口直接连接到 USART1\_RX (PA10) 和 USART1\_TX (PA9) 引脚。

注：USART1\_CK、USART1\_CTS 和 USART1\_RTS 引脚未使用，因此用户可将这些引脚用于其它外设或 GPIO。

6.3 自举程序选择

进入系统存储器自举模式且微控制器已按上述要求配置后，自举程序代码立即开始扫描 USART1\_RX 引脚，等待接收 0x7F 数据帧：一个起始位、0x7F 数据位、偶校验位和一个停止位。

此数据帧的起始位用于判断使用哪一组 USART1。此数据帧的数据位则用于串口的自动波特率检测。

随后，代码将相应初始化串行接口。通过计算出的波特率，向主机返回确认字节 (0x79)，指示 PY32 已准备好接收用户命令。

6.4 自举程序版本

下表列出了 PY32F040 器件的自举程序版本。

表 6.4-1. 自举程序版本

自举程序版本号	说明
V1.0	初始自举程序版本。

## 7 PY32F07x、PY32M07x 器件自举程序

### 7.1 自举程序配置

PY32F07x、PY32M07x 器件嵌入式自举程序支持四种串行外设：USART1、USART2、USART3、USART4、I2C 和 DFU(USB)。

下表介绍了系统存储器自举模式下 PY32F07x、PY32M07x 器件的自举程序需要使用的硬件资源。

表 7.1-1.系统存储器自举模式下 PY32 器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	时钟源	HSI 使能	使用 PLL 作为系统时钟，频率为 48MHz
	RAM	-	自地址 0x20000000 起的 2048 个字节空间供自举程序使用。
	系统存储器	-	自地址 0x1FFF0000 起的 12KB 空间包含自举程序固件。
	IWDG	-	独立看门狗(IWDG)预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件 IWDG 选项）。
USART1 自举程序	USART1	使能	初始化后，USART1 的配置为:8 位，偶校验位和 1 个停止位。
	USART1_RX 引脚	输入	PA10 引脚：USART1 用于接收。
	USART1_TX 引脚	推挽输出	PA9 引脚：USART1 用于发送。
	USART1 自动波率检测	使能	用于自动检测主机串口波特率。
	检测阶段期间，USART2_RX(PA15)、USART3_RX(PB11)、USART4_RX(PC11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
USART2 自举程序	USART2	使能	初始化后，USART2 的配置为:8 位，偶校验位和 1 个停止位。
	USART2_RX 引脚	输入	PA15 引脚：USART2 用于接收。
	USART2_TX 引脚	推挽输出	PA14 引脚：USART2 用于发送。
	USART2 自动波率检测	使能	用于自动检测主机串口波特率。
	检测阶段期间，USART1_RX(PA10)、USART3_RX(PB11)、USART4_RX(PC11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
USART3 自举程序	USART3	使能	初始化后，USART3 的配置为:8 位，偶校验位和 1 个停止位。
	USART3_RX 引脚	输入	PB11 引脚：USART3 用于接收。
	USART3_TX 引脚	推挽输出	PB10 引脚：USART3 用于发送。
	USART3 自动波率检测	使能	用于自动检测主机串口波特率。
	检测阶段期间，USART1_RX(PA10)、USART2_RX(PA15)、USART4_RX(PC11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
USART4 自举程序	USART4	使能	初始化后，USART4 的配置为:8 位，偶校验位和 1 个停止位。
	USART4_RX 引脚	输入	PC11 引脚：USART4 用于接收。
	USART4_TX 引脚	推挽输出	PC10 引脚：USART4 用于发送。
	USART4 自动波率检测	使能	用于自动检测主机串口波特率。
	检测阶段期间，USART1_RX(PA10)、USART2_RX(PA15)、USART3_RX(PB11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
I2C1 自举程序	I2C1	使能	PLL 作为 I2C 时钟源。初始化后，I2C1 的配置为：7 位地址，从模式，从地址：0x76。
	I2C1_SCL 引脚	输入/输出	PB6 引脚：I2C1 时钟线在开漏模式下使用。
	I2C1_SDA 引脚	输入/输出	PB7 引脚：I2C1 时钟线在开漏模式下使用。

	检测阶段期间，USART1_RX(PA10)、USART2_RX(PA15)、USART3_RX(PB11)、USART4_RX(PC11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
DFU 自举程序	USB_FS	使能	USBD
	USB_DM	复用	PA11: USB 发送接收数据线。
	USB_DP	复用	PA12: USB 发送接收数据线。
	中断	使能	使能 USB 中断向量以用于 USB DFU 通信。
	检测阶段期间，USART1_RX(PA10)、USART2_RX(PA15)、USART3_RX(PB11)、USART4_RX(PC11) 引脚必须保持高电平或低电平。		

系统时钟由嵌入式内部高速 RC 提供，自举程序代码不需要外部石英时钟。

下载应用程序二进制代码后，如果选择执行 Go 命令，则在跳转到用户应用程序之前，自举程序使用的外设寄存器（如上表所示）不会初始化为默认复位值。如果要使用这些寄存器，应在用户应用程序中对其重新配置。因此，如果应用程序正在使用 IWDG，则必须调整 IWDG 预分频器值来满足应用程序的要求（因为自举程序已将预分频器设置为最大值）。

7.2 自举程序硬件要求

将 PY32 置于系统存储器自举模式所需的硬件包含任何相关电路、开关或跳线，它们能够在复位期间使 BOOT0 引脚保持高电平，使 BOOT1 引脚保持低电平。  
要在系统存储器自举模式下连接 PY32，必须将 RS232 串行接口直接连接到 USART1\_RX (PA10) 和 USART1\_TX (PA9) 引脚。

注：USART1\_CK、USART1\_CTS 和 USART1\_RTS 引脚未使用，因此用户可将这些引脚用于其它外设或 GPIO。

7.3 自举程序选择

进入系统存储器自举模式且微控制器已按上述要求配置后，自举程序代码立即开始扫描 USART1\_RX 引脚，等待接收 0x7F 数据帧：一个起始位、0x7F 数据位、偶校验位和一个停止位。

此数据帧的起始位用于判断使用哪一组 USART1。此数据帧的数据位则用于串口的自动波特率检测。随后，代码将相应初始化串行接口。通过计算出的波特率，向主机返回确认字节 (0x79)，指示 PY32 已准备好接收用户命令。

7.4 自举程序版本

下表列出了 PY32F07x、PY32M07x 器件的自举程序版本。

表 7.4-1. 自举程序版本

自举程序版本号	说明
V1.0	初始自举程序版本。



## 8 PY32F403 器件自举程序

### 8.1 自举程序配置

PY32F403 器件嵌入式自举程序支持四种串行外设：USART1、USART2、USART3、USART4、I2C 和 DFU(USB)。

下表介绍了系统存储器自举模式下 PY32F403 器件的自举程序需要使用的硬件资源。

表 8.1-1.系统存储器自举模式下 PY32 器件的配置

自举程序	功能/外设	状态	备注
适用于所有自举程序	时钟源	HSI 使能	使用 PLL 作为系统时钟，频率为 48MHz
	RAM	-	自地址 0x20000000 起的 2048 个字节空间供自举程序使用。
	系统存储器	-	自地址 0x1FFF0000 起的 20KB 空间包含自举程序固件。
	IWDG	-	独立看门狗(IWDG)预分频器配置为最大值，并定期进行刷新以防止看门狗复位（如果用户事先使能硬件 IWDG 选项）。
USART1 自举程序	USART1	使能	初始化后，USART1 的配置为:8 位，偶校验位和 1 个停止位。
	USART1_RX 引脚	输入	PA10 引脚：USART1 用于接收。
	USART1_TX 引脚	推挽输出	PA9 引脚：USART1 用于发送。
	USART1 自动波率检测	使能	用于自动检测主机串口波特率。
	检测阶段期间，USART2_RX(PD6)、USART3_RX(PB11)、USART4_RX(PC11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
USART2 自举程序	USART2	使能	初始化后，USART2 的配置为:8 位，偶校验位和 1 个停止位。
	USART2_RX 引脚	输入	PD6 引脚：USART2 用于接收。
	USART2_TX 引脚	推挽输出	PD5 引脚：USART2 用于发送。
	USART2 自动波率检测	使能	用于自动检测主机串口波特率。
	检测阶段期间，USART1_RX(PA10)、USART3_RX(PB11)、USART4_RX(PC11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
USART3 自举程序	USART3	使能	初始化后，USART3 的配置为:8 位，偶校验位和 1 个停止位。
	USART3_RX 引脚	输入	PB11 引脚：USART3 用于接收。
	USART3_TX 引脚	推挽输出	PB10 引脚：USART3 用于发送。
	USART3 自动波率检测	使能	用于自动检测主机串口波特率。
	检测阶段期间，USART1_RX(PA10)、USART2_RX(PD6)、USART4_RX(PC11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
USART4 自举程序	USART4	使能	初始化后，USART4 的配置为:8 位，偶校验位和 1 个停止位。
	USART4_RX 引脚	输入	PC11 引脚：USART4 用于接收。
	USART4_TX 引脚	推挽输出	PC10 引脚：USART4 用于发送。
	USART4 自动波率检测	使能	用于自动检测主机串口波特率。
	检测阶段期间，USART1_RX(PA10)、USART2_RX(PD6)、USART3_RX(PB11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
I2C1 自举程序	I2C1	使能	PLL 作为 I2C 时钟源。初始化后，I2C1 的配置为：7 位地址，从模式，从地址：0x7E。
	I2C1_SCL 引脚	输入/输出	PB6 引脚：I2C1 时钟线在开漏模式下使用。
	I2C1_SDA 引脚	输入/输出	PB7 引脚：I2C1 时钟线在开漏模式下使用。

	检测阶段期间，USART1_RX(PA10)、USART2_RX(PD6)、USART3_RX(PB11)、USART4_RX(PC11)、USB_DM(PA11) 和 USB_DP(PA12)引脚必须保持高电平或低电平。		
DFU 自举程序	USB_FS	使能	USBD
	USB_DM	复用	PA11: USB 发送接收数据线。
	USB_DP	复用	PA12: USB 发送接收数据线。
	中断	使能	使能 USB 中断向量以用于 USB DFU 通信。
	检测阶段期间，USART1_RX(PA10)、USART2_RX(PD6)、USART3_RX(PB11)、USART4_RX(PC11)引脚必须保持高电平或低电平。		

系统时钟由嵌入式内部高速 RC 提供，自举程序代码不需要外部石英时钟。

下载应用程序二进制代码后，如果选择执行 Go 命令，则在跳转到用户应用程序之前，自举程序使用的外设寄存器（如上表所示）不会初始化为默认复位值。如果要使用这些寄存器，应在用户应用程序中对其重新配置。因此，如果应用程序正在使用 IWDG，则必须调整 IWDG 预分频器值来满足应用程序的要求（因为自举程序已将预分频器设置为最大值）。

8.2 自举程序硬件要求

将 PY32 置于系统存储器自举模式所需的硬件包含任何相关电路、开关或跳线，它们能够在复位期间使 BOOT0 引脚保持高电平，使 BOOT1 引脚保持低电平。

要在系统存储器自举模式下连接 PY32，必须将 RS232 串行接口直接连接到 USART1\_RX (PA10) 和 USART1\_TX (PA9) 引脚。

注：USART1\_CK、USART1\_CTS 和 USART1\_RTS 引脚未使用，因此用户可将这些引脚用于其它外设或 GPIO。

8.3 自举程序选择

进入系统存储器自举模式且微控制器已按上述要求配置后，自举程序代码立即开始扫描 USART1\_RX 引脚，等待接收 0x7F 数据帧：一个起始位、0x7F 数据位、偶校验位和一个停止位。

此数据帧的起始位用于判断使用哪一组 USART1。此数据帧的数据位则用于串口的自动波特率检测。

随后，代码将相应初始化串行接口。通过计算出的波特率，向主机返回确认字节 (0x79)，指示 PY32 已准备好接收用户命令。

8.4 自举程序版本

下表列出了 PY32F403 器件的自举程序版本。

表 8.4-1. 自举程序版本

自举程序版本号	说明
V1.0	初始自举程序版本。



9 版本历史

版本	日期	更新记录
V1.0	2023-07-07	初版



Puya Semiconductor Co., Ltd.

**IMPORTANT NOTICE**

Puya Semiconductor reserves the right to make changes without further notice to any products or specifications herein. Puya Semiconductor does not assume any responsibility for use of any its products for any particular purpose, nor does Puya Semiconductor assume any liability arising out of the application or use of any its products or circuits. Puya Semiconductor does not convey any license under its patent rights or other rights nor the rights of others.