



### 前言

本应用笔记讲解了当使用基于 cortex-M3 微控制器的 medium-density STM32F103xx 系列产品驱动双极步进电机时，怎样做到体积小、速度高、成本低、资源少。它提供一种简单的方法，实现全步进和半步进工作模式控制步进电机。

步进电机是一种机电设备，可将电脉冲转换为离散的机械步进运动。当以适当顺序施加电命令脉冲时，步进电机的轴将以离散步进旋转。当需要受控的运动时，步进电机是一个很好的选择。尤其是当应用需要控制旋转角、速度、位置和同步时，它们非常有用。

步进电机的主要优势在于它们不需要反馈器件；与其它运动控制系统相比，它们很便宜；它们有极佳的低速扭矩且很稳定。很多步进电机应用都得益于 STM32F10xxx 器件的功耗、特性及灵活性。包括机器人控制器、车床工具、摄影机及其它精确的轴位置控制环境。

此外，STM32F10xxx 微控制器的高性能为设计者们提供了仅需控制器的很低计算资源，就可以稳定控制步进电机。

本应用笔记给出了一个简单的方法来实现按照典型运行曲线运行（先加速，再匀速，最后减速）的步进电机控制。用户可选择工作模式（全步进或半步进）、电机的旋转方向（顺时针或逆时针）、控制电流模式（快衰减或慢衰减）。此方法使用 medium-density STM32F103xx 及 L6208 全集成两相步进电机驱动。此方法价格最低、实现最简单、使用的 CPU 负荷最小。

# 目录

<b>1</b>	<b>步进电机基础</b> .....	<b>5</b>
1.1	步进电机类型 .....	5
1.2	驱动信号 .....	6
<b>2</b>	<b>使用中密度 STM32F103xx 驱动双极步进电机</b> .....	<b>7</b>
2.1	硬件开发 .....	7
2.1.1	STM32F10xxx 用于驱动双极步进电机的特性 .....	7
2.1.2	双极步进电机的 L6208 DMOS 驱动 .....	8
2.2	固件开发 .....	13
2.2.1	固件说明 .....	13
2.2.2	源文件 .....	14
2.2.3	主程序和程序流程图 .....	14
2.2.4	步进电机驱动软件库说明 .....	16
2.2.5	固件性能研究 .....	19
<b>3</b>	<b>结论</b> .....	<b>20</b>
<b>4</b>	<b>修订历史</b> .....	<b>21</b>

## 表格索引

表 1.	L6208 的电气特性.....	10
表 2.	源文件列表.....	14
表 3.	步进电机库函数.....	17
表 4.	Stepper_ResetDisable 函数描述.....	17
表 5.	Stepper_Start 函数描述.....	17
表 6.	Stepper_Cmd 函数描述.....	17
表 7.	Stepper_PinControlConfig 函数描述.....	18
表 8.	Stepper_SelectMode 函数描述.....	18
表 9.	Stepper_SetRotationDirection 函数描述.....	18
表 10.	Stepper_SetControlMode 函数描述.....	18
表 11.	Stepper_Init 函数描述.....	18
表 12.	外设使用.....	19
表 13.	资源需求.....	19
表 14.	文档修订历史.....	21

## 图片索引

图 1.	步进电机类型 .....	6
图 2.	典型运行曲线 .....	7
图 3.	L6208 框图 .....	9
图 4.	L6208 典型应用 .....	10
图 5.	半步进模式 .....	11
图 6.	正常驱动模式 .....	12
图 7.	波形驱动模式 .....	12
图 8.	L6208 与 STM32F10xxx 之间的连接举例 .....	13
图 9.	步进电机的时钟信号 .....	14
图 10.	主程序流程图 .....	15
图 11.	DMA 程序流程图 .....	16
图 12.	Systick 程序流程图 .....	16

# 1 步进电机基础

## 1.1 步进电机类型

有三种基本的步进电机类型：

- 可变磁阻
- 永磁
- 混合

可变磁阻（VR）电机类型已经存在了很长时间。从构造角度，它可能是最容易理解的。此类电机包含一个软铁多齿转子和一个绕线定子。当向定子绕组输入 DC 电流时，定子励磁为磁极，当转子的齿被吸到通电的定子磁极时，会发生旋转。

永磁（PM）电机类型向电机结构增加了永磁体。转子不像 VR 电机中那样有齿，而是磁化为交替的北极和南极，两极位置在与转子轴平行的直线上。这些磁化转子极提供了增强的磁通量密度，与 VR 类型相比，它为 PM 电机带来了更好的扭矩特性。

混合（HB）电机类型在步进分辨率、扭矩和速度方面有更好的性能。此类电机融合了 PM 和 VR 步进电机这两种类型的最佳特性。与在 VR 电机中一样，转子是多齿的，在其转轴周围包含轴向磁化的同心磁。转子上的齿提供了更好的路径，有助于将磁通引导至气隙中的所需位置。与 VR 和 PM 电机类型相比，此特性增加了电机的制动、保持和动态磁通特性。

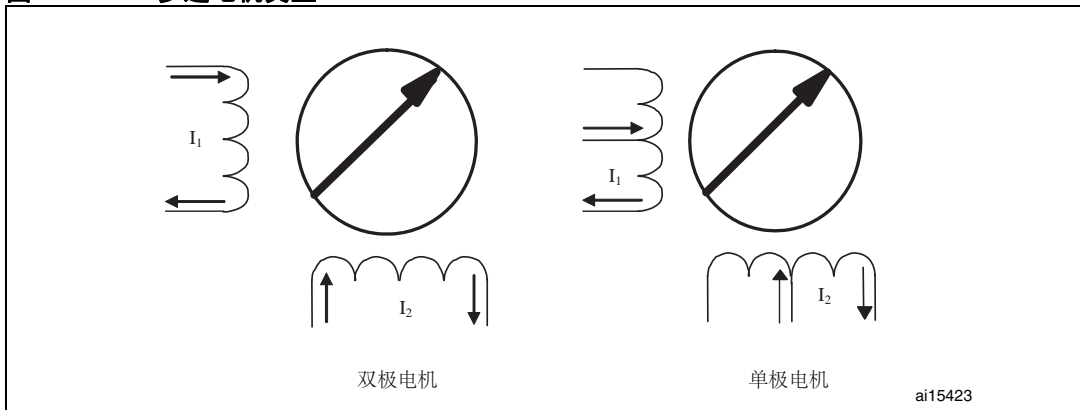
步进电机可为双极或单极绕组。

单极步进电机有两个相同的线圈，它们不做电连接，都有中心抽头。双线无感线圈组的任一端通电，共用中心抽头，则磁通反转。单极步进电机的优势在于它们仅需要一个切换开关。然而，它们需要双线无感线圈组，这意味着对于一个给定的体积因子，线更细，电阻更高。

除了线圈没有中心抽头外，双极步进电机与单极电机相同。对于双极电机，可使用 H 桥反转绕组的极性及磁通。双极步进电机的优势在于它们仅使用一个线圈，因此具有较好的体积因子（较低的线圈电阻）。

单极电机仍被广泛使用，因为当与分立器件一起实现时，它们的驱动电路更简单。然而，对于现今的集成电路，驱动双极电机所需的元件也不比单极电机多。

图 1. 步进电机类型



## 1.2 驱动信号

当供电时，直流电机可自行运行，而步进电机需要换向信号。

可使用不同模式驱动步进电机，包括全步进和半步进模式。通常全步进模式被全步进电机驱动使用。在此模式中，两个相位都一直被供电，电机具有全额定扭矩。此控制模式仅需四个矩形信号，也可由 PWM 生成（在一步的时间内有固定的占空比）。取决于超前相位，电机轴会顺时针或逆时针旋转。

半步进模式更复杂一些。若使用了半步进驱动，则在每个时钟沿后，电机前进半步，这样能得到更高的位置分辨率及降低不稳定性。

然而，在两种模式中，信号都以确定的方式互相相关，因此可用标准逻辑生成。较好的逻辑实现最好使用专业应用集成电路，当然芯片价格也很高。简单讲，专业应用集成电路包含内部转换电路，由步长和方向输入控制。IC（集成电路）电机控制器仅有三种工作模式：全步进两相开、半步进、波形驱动。此类 IC 需要输入四个信号至控制器，它们由微电脑或其它专用控制芯片提供。此外，此解决方案需要一个功率驱动电路和一个微处理器，来产生不同电机的驱动控制信号。

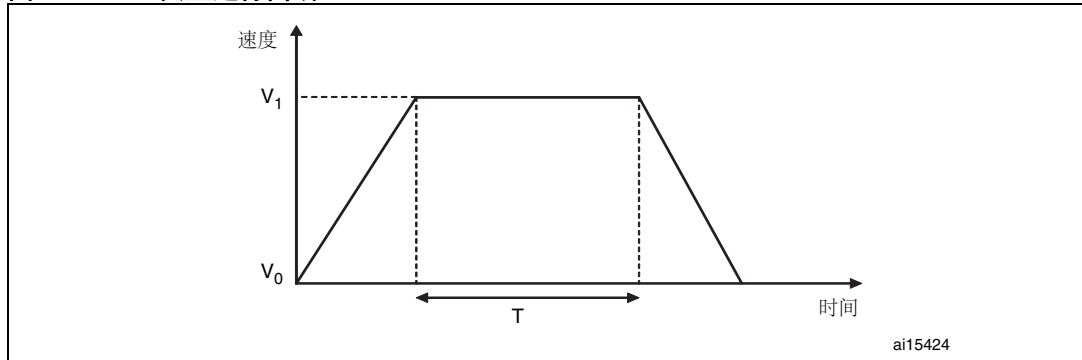
一些已知应用需要能改变步进电机速度，以生成确定的速度曲线（例如，一个典型的运行曲线）。在此情况下，在这三种可能的模式中，驱动步进电机的最好方式为软件解决方案，使用能代替任何其它控制器的微控制器电路。在基于微控制器的应用中，可使用软件和 PWM 定时器，因此不需要外部控制器。

使用此方法可用最少的硬件及非常低的微控制器 CPU 负荷实现高速应用。

## 2 使用中密度 STM32F103xx 驱动双极步进电机

本章说明了如何根据图2中所示的典型运行曲线，使用 medium-density STM32F103xx 来驱动全步进和半步进模式的步进电机。

图 2. 典型运行曲线



根据图2，步进电机从速度  $V_0$  加速至速度  $V_1$ ，在此速度保持时间  $T$ 。之后，电机减速至初始速度  $V_0$ 。速度曲线在加速和减速期间的斜率相同。

### 2.1 硬件开发

如前面章节所述，需要正确选择微控制器和 IC，以实现驱动步进电机的最佳解决方案。在本应用中，选择的是 medium-density STM32F103xx 器件和 L6208 DMOS 驱动。

#### 2.1.1 STM32F10xxx 用于驱动双极步进电机的特性

medium-density STM32F103xx 的外设非常适合驱动步进电机。这些外设包括三个标准（通用）定时器（TIM2、TIM3、TIM4），其内部时钟频率最高为 72 MHz，以及用于高分辨率捕获的四个 16 位独立通道。此外，STM32F103xx 外设包括一个高级控制定时器（TIM1），其内部时钟频率最高为 72 MHz，以及四个 16 位高分辨率捕获通道。此定时器有三个通道（CH1、CH2、CH3），它们可生成三个互补信号。它还有一个独立通道（CH4）。这些定时器通道可生成 PWM 中的信号或输出比较模式中的信号。要生成 L6208 控制步进电机所需的常规步进时钟输入，需要 PWM 和输出比较特性。本应用的目的是在输出比较切换模式中使用 TIM2 控制步进电机速度，占空比恒定为 50%，频率可变。后者用于改变电机速度，使之与典型速度曲线相符。当更改时钟频率时，需要最低 CPU 负荷以防止驱动的步进电机停转。

用 DMA 控制器传输定时器周期可加速 CPU 的工作，这是因为通过 DMA 控制器，周期数据直接由存储器传输到定时器控制寄存器，而不需要 CPU 干预。STM32F10xxx 的 DMA 控制器有七个独立可配通道，每个通道有三个事件标志：一个用于 DMA 半传输，一个用于 DMA 传输完成，最后一个用于 DMA 传输错误。在本应用中，使用更新事件 DMA 请求和 DMA 传输完成中断来控制周期的传输。对于所有定时器，更新事件请求都存在于不同的 DMA 通道中。因此，此选择的优势在于：所有定时器与其 DMA 请求都能同时工作以驱动不同的步进电机。


SysTick 用于在  $T$  时间内维持最大步进电机速度。每次达到所编程的时间基准时，这一灵活的系统定时器都可生成一个中断。

medium-density STM32F103xx 有八十多个 GPIO，可配置为复用功能推挽。这些 I/O 可控制步进电机的旋转方向（顺时针或逆时针）、步进模式（全步进或半步进）、衰变模式（慢或快）、L6208 芯片使能信号。（若需 medium-density STM32F103xx 特性的更详细信息，请参考 STM32F10xxx 参考手册）。

总之，使用这些外设，medium-density STM32F103xx 可同时驱动十九个全步进及半步进模式的步进电机，仅需消耗最低的 CPU 负荷。此解决方案的硬件需求为：功率极中每个线圈都要有一个集成了基本 H 桥电路的 IC，以及主要用于相位生成的中央逻辑。

## 2.1.2 双极步进电机的 L6208 DMOS 驱动

用于双极步进电机的驱动电路更复杂一些，因为双极电机的绕组上没有中心抽头。因此，若需反转电机线圈所产生场的方向，需要使用 H 桥电路，通过线圈反转电流。市场上有很多集成的 H 桥驱动。本应用选择的驱动为 L6208，它是 DMOS 全集成步进电机驱动，其非耗散过电流保护用于驱动两相双极步进电机。它包含一个双 DMOS 全桥、用于执行浪涌调节的两个固定关断时间 PWM 电流控制器（每个桥一个）、用于生成步进序列的一个相位序列生成器。

 表示了 L6208 IC 框图。对于此图，主要的 L6208 输入为：

- **EN 逻辑输入：**它是芯片使能输入。低逻辑电平可切断桥 A 和桥 B 的所有功率 MOSFET。此引脚还连至过电流和热保护电路的集电极以实现过电流保护。
- **CLOCK 逻辑输入：**它是步进时钟输入。在每个上升沿，状态机前进一步。
- **CW/CCW 逻辑输入：**它选择了旋转的方向。高逻辑电平设置的是顺时针方向，低逻辑电平设置的是逆时针方向。
- **HALF/FULL 逻辑输入：**它是步进模式选择器。高逻辑电平设置的是半步进模式，低逻辑电平设置的是全步进模式。
- **CONTROL 逻辑输入：**它是衰变模式选择器。高逻辑电平设置的是慢衰变模式，低逻辑电平设置的是快衰变模式。
- **VREF<sub>A</sub> 和 VREF<sub>B</sub> 模拟输入：**它们分别是桥 A 和桥 B 的源引脚。



图 3. L6208 框图

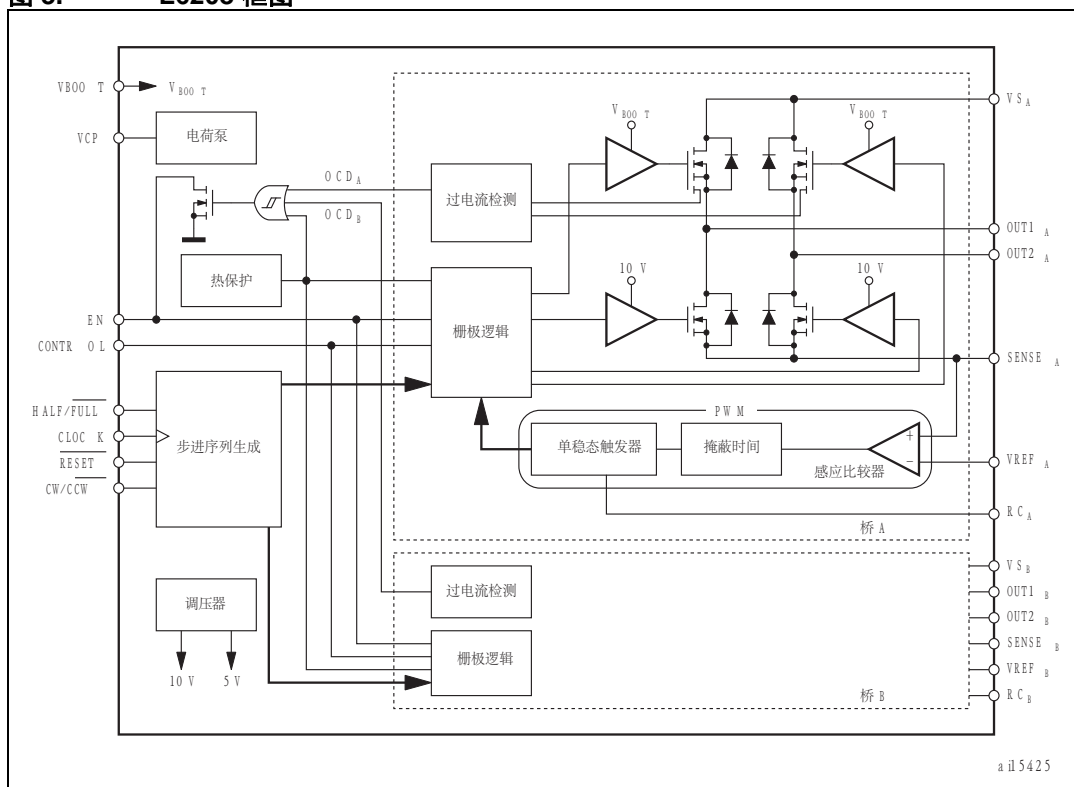


图 4 显示了使用 L6208 驱动的典型双极步进电机驱动应用。它还显示了所用封装，即 PowerDIP24/SO24 的不同引脚。在功率引脚 ( $V_{SA}$  和  $V_{SB}$ ) 和 L6208 附近的接地之间应插入范围为 100 到 200 nF 的高质量陶瓷电容，以增强电源的高频滤波，降低开关产生的高频瞬态。EN 输入和接地之间所连接的电容可设置检测到过电流时的关闭时间。两个电流感应输入 ( $SENSE_A$  和  $SENSE_B$ ) 应与布局中的感应电阻尽量接近。感应电阻应是非电感电阻，以使通过它的  $di/dt$  瞬态最低。为增强对噪声的免疫力，最好将不用的逻辑引脚（除 EN 之外）连至 5 V（高逻辑电平）或 GND（低逻辑电平）。建议在 PCB 上将电源地和信号地分开。

图 4. L6208 典型应用

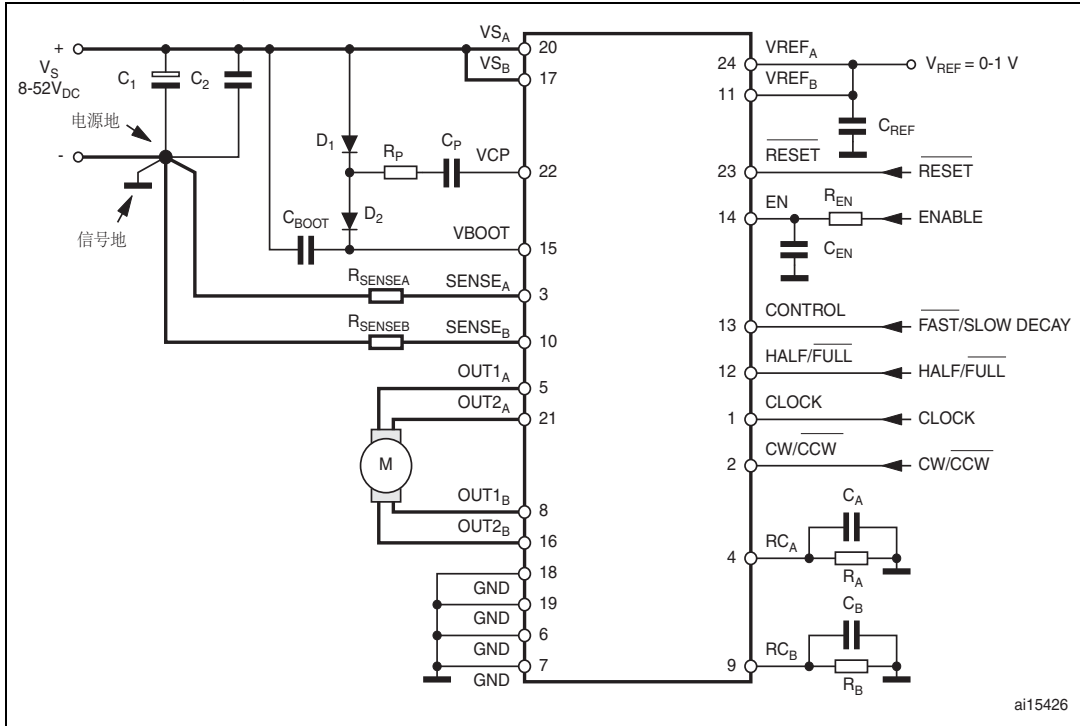


表 1 给出了 L6208 驱动的电气特性。

表 1. L6208 的电气特性

参数	最小值	最大值	单位
电源电压 (VS)	8	52	V
输出电流 (IOUT)		2.8	A
开关频率 (fSW)		100	kHz
工作结温 (Tj)	-25	+125	°C

L6208 两个桥中的每个桥都包括一个固定的关断时间 PWM 电流控制器。通过感应两个低功率 MOS 晶体管的源和地之间所连外部感应电阻上的压降，电流控制电路取得桥电流。

每个电机线圈的电流流过相应的感应电阻，产生压降，逻辑电路可用其控制负载电流的峰值。当选择感应电阻 RSENSE 的值时，必须考虑两个问题：

- 当电流再开始流动时，感应电阻耗散功率会在 SENSE 引脚上引起潜在的危险负电压。因此，该元件的电阻应较低。
- 由内部比较器比较 RSENSE 上的压降与 (Vref 引脚上的) 参考电压。RSENSE 值越低，峰值电流误差就越高。这是因为 Vref 引脚上的噪声以及电流感应比较器的输入偏置：必须避免太小的 RSENSE 值。

计算感应电阻值的一个较好折中为，对应于流过负载峰值电流 (Ipeak) 的压降应为约 0.5 V：

$$R_{SENSE} = 0.5 \text{ V} / I_{peak}$$

感应电阻必须为非电感类型，以避免 SENSE 引脚上有危险的负电压。

### 电流控制模式：快衰变模式和慢衰变模式

CONTROL 输入用于选择关断期间桥的行为。当 CONTROL 引脚为低时，选择的是快衰变模式，桥中的两个晶体管都在关断期间关闭。当 CONTROL 引脚为高时，选择的是慢衰变模式，仅桥的低侧晶体管在关断期间关闭。

### 步进序列生成：半步进和全步进模式

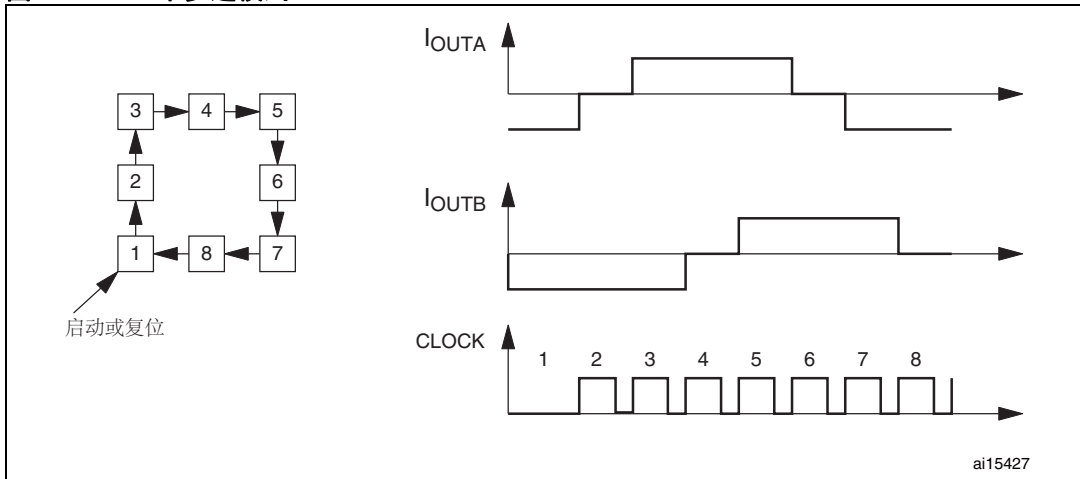
相位序列生成器为一个状态机，它可提供相位并启用两个桥的输入，以在全步进或半步进模式中驱动步进电机。有两种全步进模式：普通驱动模式，在每步时两个相位都开；波形驱动模式，同一时间仅有一个相位为开。驱动模式由 HALF/FULL 输入选择。CLOCK 输入的上升沿会使状态机前进至下一状态。旋转的方向由 CW/CCW 输入设置。RESET 输入会将状态机复位至 Home 状态（状态 1）。

#### ● 半步进模式

HALF/FULL 输入上的高逻辑电平选择的是半步进模式。图 5 显示了电机电流波形及相位序列生成器的状态图。在启动时或复位后，相位序列生成器处于状态 1（Home 状态）。在每个时钟脉冲之后，状态根据如下顺序改变：

- 1,2,3,4,5,6,7,8,... 若 CW/CCW 为高（顺时针方向）
- 1,8,7,6,5,4,3,2,... 若 CW/CCW 为低（逆时针方向）。

图 5. 半步进模式

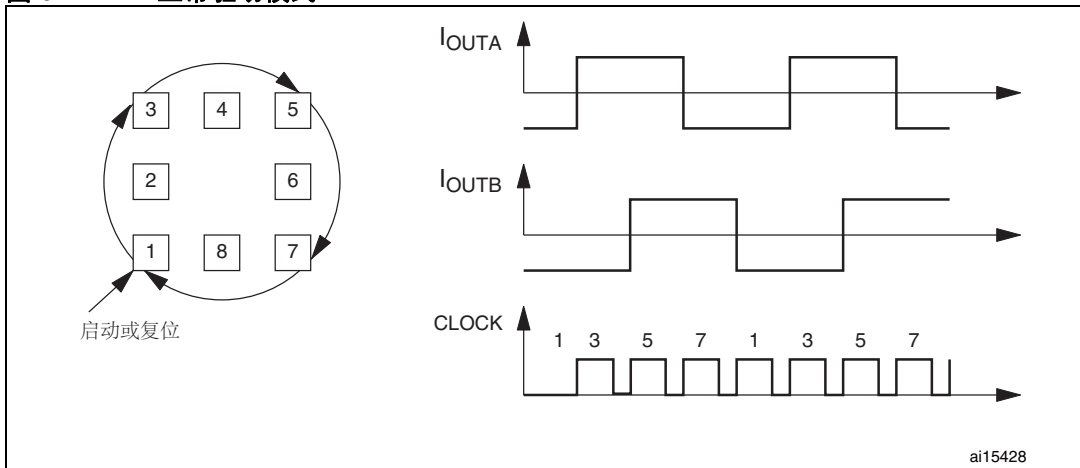


#### ● 普通驱动模式（全步进，两相开）

HALF/FULL 输入上的低电平选择的是全步进模式。若为低电平，当状态机为奇数状态时，选择普通驱动模式。图 6 显示了电机电流波形及相位序列生成器的状态图。保持 HALF/FULL 输入为低然后复位，即可选择普通驱动模式。在启动时或复位后，状态机处于状态 1。当 HALF/FULL 输入保持为低时，状态按如下顺序变化：

- 1,3,5,7,... 若 CW/CCW 为高（顺时针方向）
- 1,7,5,3,... 若 CW/CCW 为低（逆时针方向）

图 6. 正常驱动模式



● 波形驱动模式（全步进，一相开）

HALF/FULL 输入引脚上的低电平选择的是全步进模式。若为低电平，当状态机为偶数状态时，选择波形驱动模式。[图 7](#) 显示了电机电流波形及相位序列生成器的状态图。要进入波形驱动模式，状态机必须为偶数状态。更直接进入波形驱动模式的方法为：首先复位，然后在保持 HALF/FULL 输入为高电平的同时向时钟输入施加一个脉冲，然后将 HALF/FULL 输入变为低电平。此顺序首先将状态机强制为状态 1。取决于 CW/CCW 输入，时钟脉冲与 HALF/FULL 输入高电平共同将状态机从状态 1 切换至状态 2 或 8。在每个时钟脉冲（上升沿）之后，状态机根据如下顺序改变状态：

- 2,4,6,8,... 若 CW/CCW 为高（顺时针方向）
- 8,6,4,2,... 若 CW/CCW 为低（逆时针方向）

有关详细信息，请参见 L6208 数据手册。

图 7. 波形驱动模式

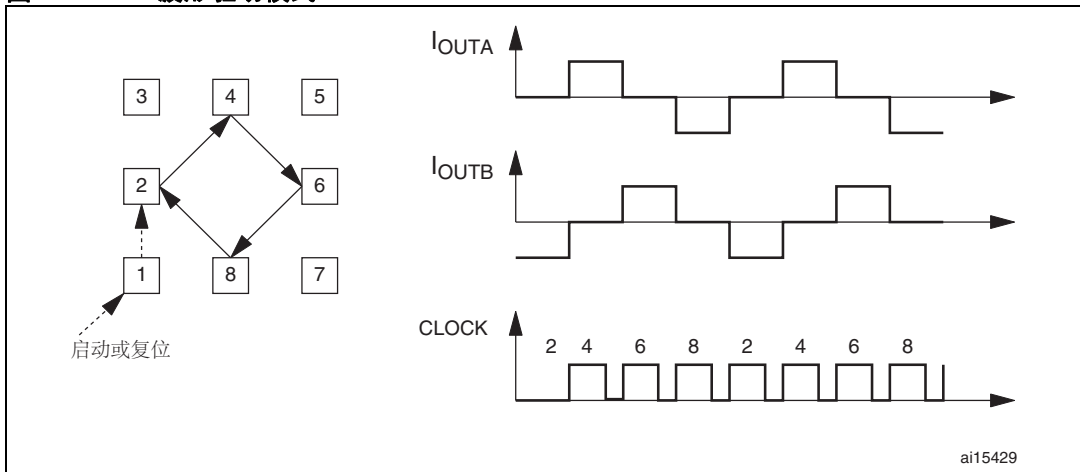
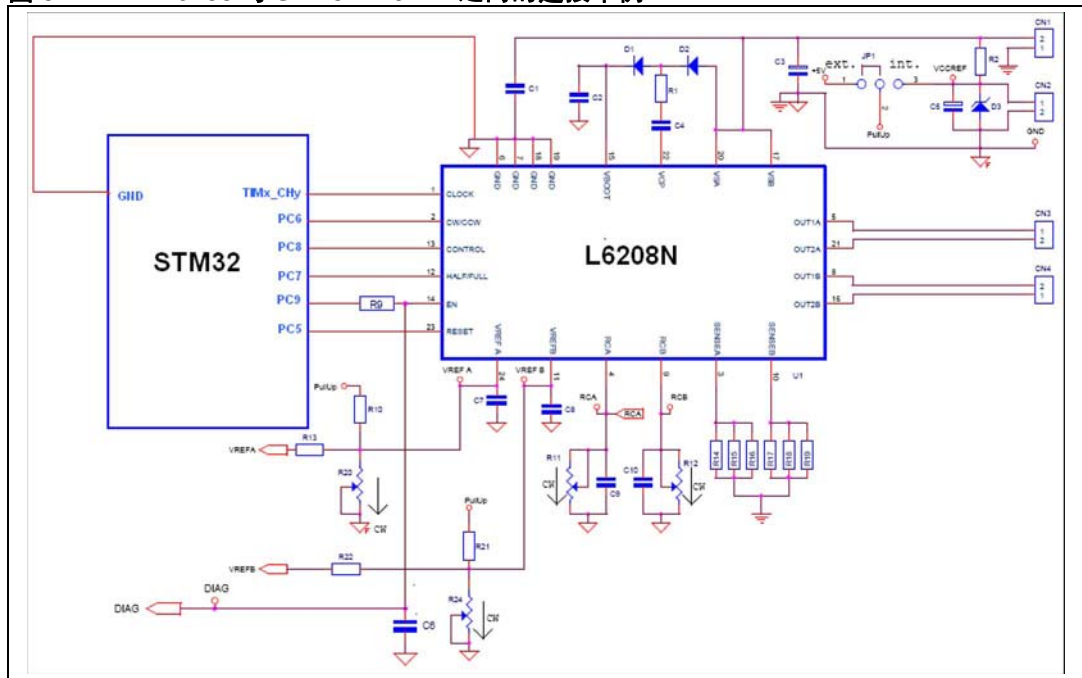


图 8. L6208 与 STM32F103xx 之间的连接举例



## 2.2 固件开发

为保证控制步进电机时的 CPU 负荷最低，一个优化的程序利用了 STM32F103xx 器件的资源 and 性能。

### 2.2.1 固件说明

在本应用中，TIM2 用于输出比较切换模式，以更改信号时钟频率。加速期间信号频率从 200 Hz 增加至 800 Hz，减速期间从 800 Hz 到 200 Hz，占空比为 50%。当定义频率和脉冲时，必须考虑 L6208 的电气特性。L6208 的最大时钟频率为 100 kHz，时钟低和高的最小时间为 1  $\mu$ s。输出比较切换模式提供了在保持占空比为常量的同时仅改变频率的可能，以使用最低 CPU 负荷得到常规时钟信号。由 DMA 控制器确保从存储器到 ARR 定时寄存器传输周期。

使用两个缓存更改 TIM2 周期。它们存储于 SRAM 中。SRC\_Buffer\_INC 为步进电机加速缓冲，SRC\_Buffer\_DEC 为电机减速缓冲。在本应用中，加速和减速的斜率相同。每个缓冲都包含输入时钟信号的十个频率。

在加速阶段，当 SRC\_Buffer\_INC 中的所有周期传输完成之后，会生成 DMA 传输完成中断。此中断会停止 DMA 传输，启用 SysTick 开始 8 ms 的计时时间 T，在此期间步进电机继续运行于最大速度。在 T 之后，会生成 SysTick 中断，启用 DMA 传输来传输 SRC\_Buffer\_DEC 中的周期。最后一个周期传输完成之后，发生 DMA 传输完成的中断。每 38 ms 重复一次整个过程。

图 9. 步进电机的时钟信号

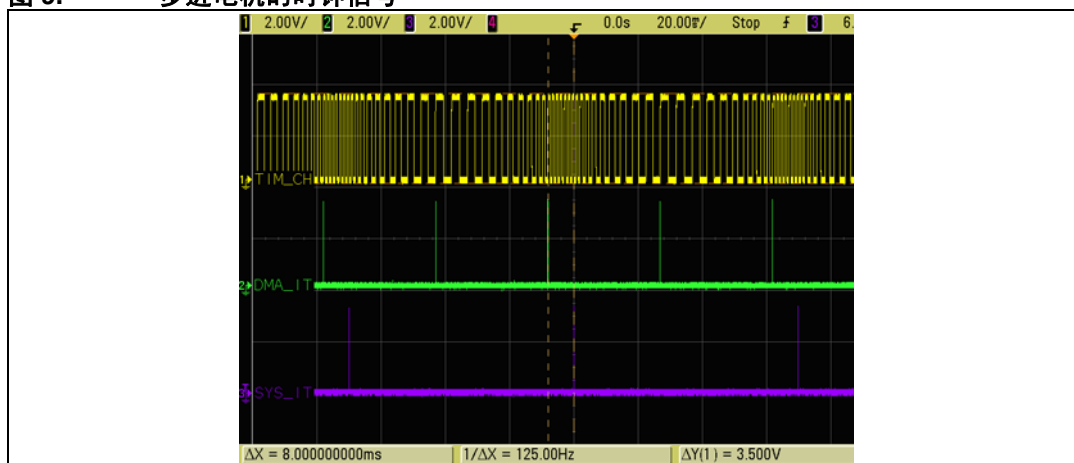


图 9 用黄色显示了为步进电机生成的时钟信号。另外还显示了 DMA 中断和 SysTick 中断。当保持最大频率（800 Hz）时，测量周期 T。

## 2.2.2 源文件

本应用笔记提供的程序所包括的源文件列于下面的表 2 中。

表 2. 源文件列表

文件	说明
StepperMotor.c	用于使用不同配置和模式控制步进电机的程序。
main.c	样例程序。
STM32f10x_it.c	中断服务程序。

## 2.2.3 主程序和程序流程图

若需测试步进电机控制的样例，应在 main 源文件中去掉一些 #define 语句的注释：

1. 去掉步进模式两个选项之一的注释：

```
#define Half_Step
    选择的模式为半步进模式。
```

```
#define Full_Step
    选择的模式为全步进模式。
```

2. 去掉电机旋转方向两个选项之一的注释：

```
#define RotationDirection_CW
    电机顺时针方向旋转。
```

```
#define RotationDirection_CCW
    电机逆时针方向旋转。
```

3. 去掉电流衰变模式两个选项之一的注释：

```
#define ControlSlow_Current
```

选择慢衰变模式。

```
#define ControlFast_Current
```

选择快衰变模式。

此外，用户应在工具选项中选择 STM3210B-EVAL 板。

图 9、图 10 和图 11 分别提供了主程序的流程图、DMA 程序和 SysTick 程序。

图 10. 主程序流程图

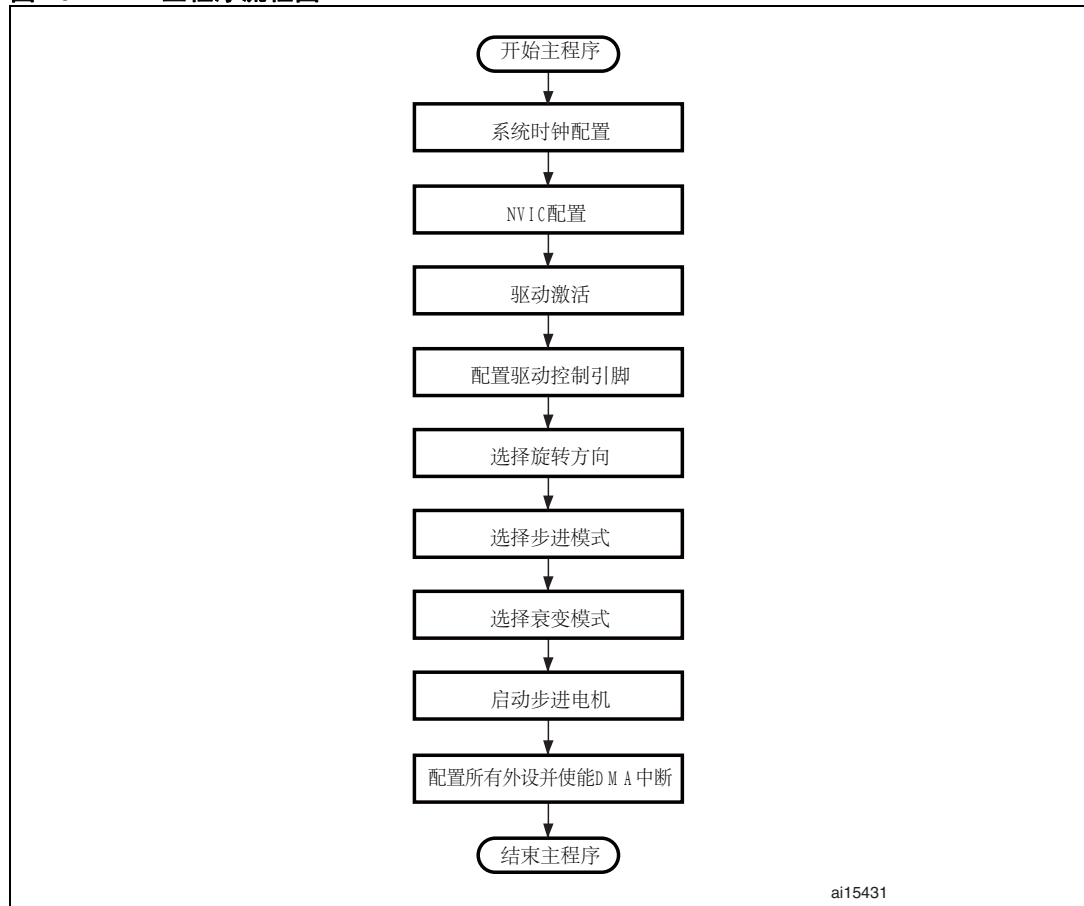


图 11. DMA 程序流程图

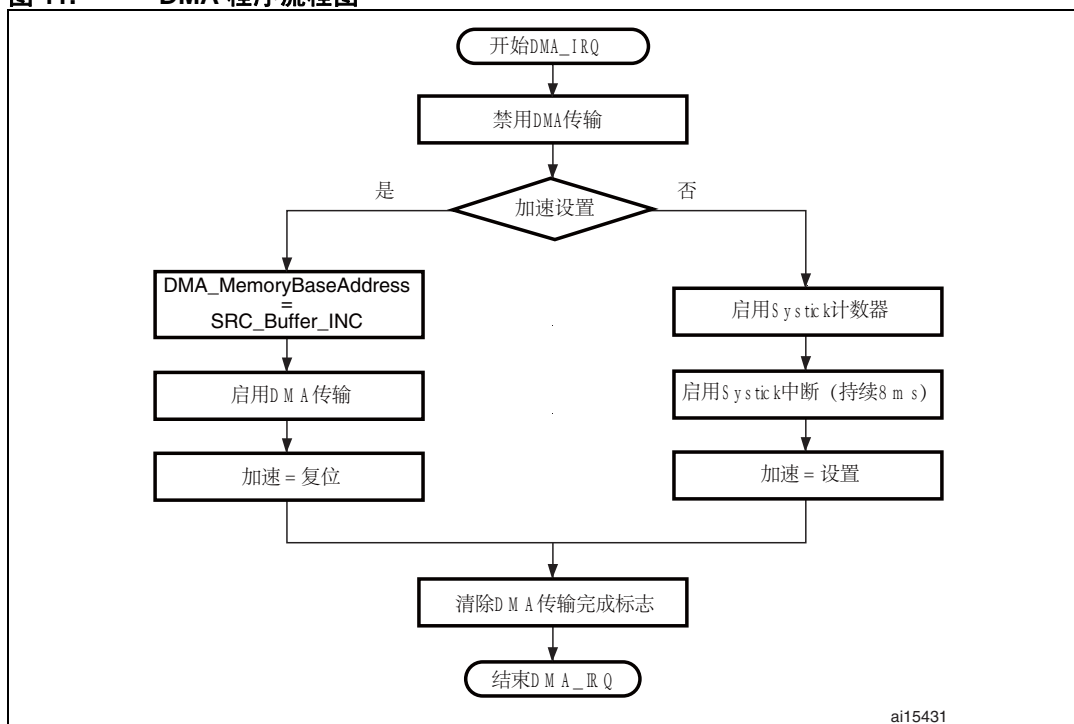
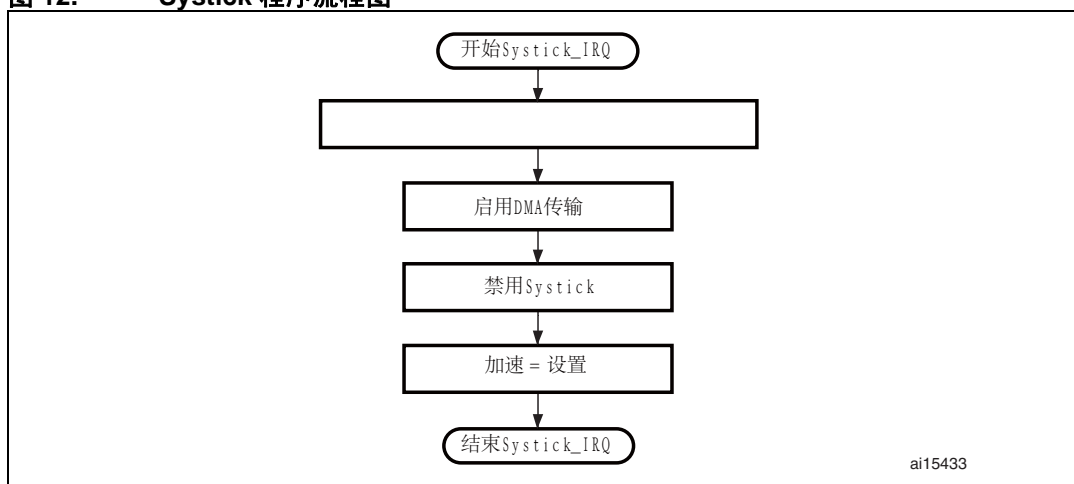


图 12. Systick 程序流程图



### 2.2.4 步进电机驱动软件库说明

本节说明了步进电机驱动的软件库，给出了相关函数的详细信息，提供了使用这些函数的样例及典型运行曲线的样例。

步进电机驱动可用于管理步进电机功能，可支持全步进及半步进模式、顺时针及逆时针旋转方向、慢衰变及快衰变模式。

表 3 列出了步进电机库的各个函数。



**表 3. 步进电机库函数**

函数名称	说明
Stepper_ResetDisable	将驱动的初始化禁用至其默认复位值。
Stepper_Start	开始或停止步进电机。
Stepper_Cmd	通过启用或禁用外设时钟，激活或取消激活驱动。
Stepper_PinControlConfig	配置驱动控制引脚。
Stepper_SelectMode	选择步进模式：全步进或半步进。
Stepper_SetRotationDirection	选择旋转方向：CW 或 CCW。
Stepper_SetControlMode	指定衰变模式：慢或快。
Stepper_Init	配置所有外设，控制步进电机，启用 DMA 中断。

Stepper\_ResetDisable 函数在 [表 4](#) 中说明。

**表 4. Stepper\_ResetDisable 函数描述**

函数名称	Stepper_ResetDisable
函数原型	void Stepper_ResetDisable(void)
行为说明	将驱动的初始化禁用至其默认复位值。
输入	无
输出	无

Stepper\_Start 函数在 [表 5](#) 中说明。

**表 5. Stepper\_Start 函数描述**

函数名称	Stepper_Start
函数原型	void Stepper_Start(FunctionalState NewState)
行为说明	开始或停止步进电机。
输入	NewState: ENABLE 或 DISABLE
输出	无

Stepper\_Cmd 函数在 [表 6](#) 中说明。

**表 6. Stepper\_Cmd 函数描述**

函数名称	Stepper_Cmd
函数原型	void Stepper_Cmd(FunctionalState NewState)
行为说明	通过启用或禁用总线时钟，激活或取消激活驱动。
输入	NewState: ENABLE 或 DISABLE
输出	无

Stepper\_PinControlConfig 函数在 [表 7](#) 中说明。

**表 7. Stepper PinControlConfig 函数描述**

函数名称	Stepper_PinControlConfig
函数原型	void Stepper_PinControlConfig(void)
行为说明	配置驱动控制引脚
输入	无
输出	无

Stepper\_SelectMode 函数在 [表 8](#) 中说明。

**表 8. Stepper SelectMode 函数描述**

函数名称	Stepper_SelectMode
函数原型	void Stepper_SelectMode(uint16_t Stepper_Mode)
行为说明	选择步进模式：全步进或半步进模式
输入	Stepper_Mode: Stepper_Full 或 Stepper_Half
输出	无

Stepper\_SetRotationDirection 函数在 [表 9](#) 中说明。

**表 9. Stepper SetRotationDirection 函数描述**

函数名称	Stepper_SetRotationDirection
函数原型	Stepper_SetRotationDirection(uint16_t Stepper_RotationDirection)
行为说明	选择旋转方向：CW 或 CCW
输入	Stepper_RotationDirection: Stepper_RotationDirection_CW 或 Stepper_RotationDirection_CCW
输出	无

Stepper\_SetControlMode 函数在 [表 10](#) 中说明。

**表 10. Stepper SetControlMode 函数描述**

函数名称	Stepper_SetControlMode
函数原型	void Stepper_SetControlMode(uint16_t Stepper_ControlMode)
行为说明	指定衰变模式。
输入	Stepper_ControlMode: Stepper_ControlFast 或 Stepper_ControlSlow
输出	无

Stepper\_Init 函数在 [表 11](#) 中说明。

**表 11. Stepper Init 函数描述**

函数名称	Stepper_Init
函数原型	void Stepper_Init(void)

表 11. Stepper Init 函数描述 (续)

行为说明	配置所有外设，控制步进电机，启用 DMA 中断。
输入	无
输出	无

## 2.2.5 固件性能研究

在表 12 中，说明了本应用中控制步进电机遵循典型运行曲线所使用的外设。表 13 指出了资源需求。

表 12. 外设使用

外设	说明	中断使能
5 个 I/O 引脚	步进电机输出引脚	
1 个 I/O 引脚	用作输出中的 TIM2 Channel1	
TIM2	在输出比较切换模式中配置	
DMA	使用更新事件请求，从存储器到定时器传输周期	传输完成中断 (Transfer complete interrupt)
Systick	在时间 T 期间保持最大电机速度	每当时间 T 到期时，生成一个中断

为控制步进电机遵循典型运行曲线，软件使用三个中断：加速期间的 DMA 中断、SysTick 中断、减速期间的另一个 DMA 中断。每 38 ms 生成一组这些中断。在此实现中，每个中断程序花费 12 个周期进入中断，12 个时钟周期退出中断。对于这三个中断，它们总共花费 72 个时钟周期。对于 72 MHz 的时钟速度，进入和退出这些中断的时间小于 1  $\mu$ s。

第一个 DMA 中断花费 3.6  $\mu$ s，sysTick 中断花费 2  $\mu$ s，第二个 DMA 中断花费 2  $\mu$ s。因此，步进电机控制花费的时间小于 8.6  $\mu$ s。若每 38 ms 需要一组中断，则步进电机处理仅使用了  $2.10^{-4}$ % 的 CPU 处理能力。

表 13. 资源需求

RAM	Flash
1.5 Kbit	7.5 Kbit

### 3 结论

在本应用中，使用 medium-density STM32F103xx 控制步进电机。用户可选择步进模式、旋转方向和电流控制模式。虽然在现今的市场上，也能比较容易地找到类似功能的器件，但在这里讨论的解决方案比专用 IC 具有更高的灵活性。STM32F10xxx 器件的一组外设效率极佳，非常适合驱动步进电机遵循典型运行曲线。此外，本应用的实现方法非常节约成本，这体现在外部元件和 CPU 使用量上，原因是应用使用了最少的硬件和非常简单灵活的固件程序。

## 4 修订历史

表 14. 文档修订历史

日期	修订	变更
2009 年 3 月 06 日	1	初始版本。
2009 年 4 月 30 日	2	根据新的 STM32F10xxx 标准外设库 (StdPeriph_Lib) V3.0.0, 将所有 u16 更新为 uint16_t。

**请仔细阅读：**

中文翻译仅为方便阅读之目的。该翻译也许不是对本文档最新版本的翻译，如有任何不同，以最新版本的英文原版文档为准。

本文中信息的提供仅与 ST 产品有关。意法半导体公司及其子公司（“ST”）保留随时对本文档及本文所述产品与服务进行变更、更正、修改或改进的权利，恕不另行通知。

所有 ST 产品均根据 ST 的销售条款出售。

买方自行负责对本文所述 ST 产品和服务的选择和使用，ST 概不承担与选择或使用本文所述 ST 产品和服务相关的任何责任。

无论之前是否有任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为 ST 授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在 ST 的销售条款中另有说明，否则，ST 对 ST 产品的使用和 / 或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

意法半导体的产品不得应用于武器。此外，意法半导体产品也不是为下列用途而设计并不得应用于下列用途：（A）对安全性有特别要求的应用，例如，生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）汽车应用或汽车环境，且 / 或（D）航天应用或航天环境。如果意法半导体产品不是为前述应用设计的，而采购商擅自将其用于前述应用，即使采购商向意法半导体发出了书面通知，采购商仍将独自承担因此导致的任何风险，意法半导体的产品规格明确指定的汽车、汽车安全或医疗工业领域专用产品除外。根据相关政府主管部门的规定，ESCC、QML 或 JAN 正式认证产品适用于航天应用。

经销的 ST 产品如有不同于本文档中提出的声明和 / 或技术特点的规定，将立即导致 ST 针对本文所述 ST 产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大 ST 的任何责任。

ST 和 ST 徽标是 ST 在各个国家或地区的商标或注册商标。

本文档中的信息取代之前提供的所有信息。

ST 徽标是意法半导体公司的注册商标。其他所有名称是其各自所有者的财产。

© 2015 STMicroelectronics 保留所有权利

意法半导体集团公司

澳大利亚 - 比利时 - 巴西 - 加拿大 - 中国 - 捷克共和国 - 芬兰 - 法国 - 德国 - 中国香港 - 印度 - 以色列 - 意大利 - 日本 - 马来西亚 - 马耳他 - 摩洛哥 - 菲律宾 - 新加坡 - 西班牙 - 瑞典 - 瑞士 - 英国 - 美国

www.st.com