

EM9316B 数据采集卡编程手册

北京中泰联创科技有限公司

版权信息

本软件产品及相关套件版权均属北京中泰联创科技有限公司所有, 您若需要我公司产品及相关信息请及时与当地代理商或直接与我们联系, 我们将热情接待。

目 录

第一章 名词解释	3
第二章 使用动态链接库编程方法	3
2.1 概述	3
2.2 模拟量（AD）编程方法	4
2.3 开关量编程方法	6
2.4 编码器编程方法	8
2.5 计数器编程方法	9
2.6 PWM 编程方法	11
2.7 DA 编程方法	14
2.8 高速同步采集编程方法	15
2.9 读取离线文件编程方法	18
第三章 函数原型与功能详解	20
3.1 数据类型	20
3.2 函数说明	20
第四章 基本原理	20
4.1 物理值转换与数据格式	20
4.2 计数器测频原理	22
4.3 PWM 频率输出原理	23
更新记录	24

第一章 名词解释

下位机：因为 EM9316 本身具有一定运算能力，可以看作是一个独立的计算机，所以称为下位机
上位机：指和 EM9316 通讯的计算机。

第二章 使用动态链接库编程方法

2.1 概述

EM9316 是 16 位 100KHz 多功能并行采集设备，板载 64MB 硬件缓冲区保证其可以长时间连续采集。如果使用网络接口，无需安装驱动即可操作设备。如果使用 USB 接口，则需要首先安装驱动才能够使用设备。

EM9316 工作在 EM9636 兼容模式下时，可以直接使用 EM9636 的通讯协议进行编程，这可以让 EM9636 的用户无需更改自己的测试程序直接使用 EM9316，如果用户之前没有用过 EM9636，则请阅读本手册，使用 EM9316 专有的编程方法，从而充分发挥 EM9316 的硬件功能。

在使用本函数库时，请确保“EM9636.dll”至少在下面目录中的一个：

1. 工程目录
2. 可执行文件目录
3. 系统目录

如果仍然发生无法找到动态链接库的错误，则请联系我公司技术支持，我们将根据具体情况提供解决方法。

1.2 节到 1.6 节是单项功能的编程方法，1.7 节介绍了 AD、编码器、计数器全部都需要高速同步采集时的编程方法。

2.2 模拟量（AD）编程方法

2.2.1 相关函数：

EM9316_DeviceCreate
EM9316_DeviceClose
EM9316_CmdConnect
EM9316_CmdClose
EM9316_SfifoReadData
EM9316_SFifoCanReadBC
EM9316_HcFifoStatus
EM9316_AdChGetCode
EM9316_AdChCodeToValue
EM9316_AdGetMaxChCnt
EM9316_AdChIsInFifo
EM9316_HcSetGroupFreq
EM9316_TriSetSrc
EM9316_TriSetMode
EM9316_HcSetGroupCnt
EM9316_AdSetRange
EM9316_HcSetOversampleInx

2.2.2 内部定时采集：

本设备板载 64MB 缓存，在启动定时采集后，下位机先将数据存储在缓冲区中，上位机 DLL 开辟上位机缓冲区，自动将数据读出放到上位机缓冲区中。用户可以调用 EM9316_SfifoReadData 函数读回上位机缓冲区中的数据。缓冲区是 FIFO 形式，数据读出后就不再保存在缓冲区中。

关于 EM9316_SfifoReadData 函数的使用，有下面几个建议：

1.每次读出的数据字节数（byteCount）为缓冲区中每组字节数的整数倍，这样便于后续的数据处理。可以通过 EM9316_GetFifoGroupByteCount 函数得到每组字节数。

2.超时时间 timeOutMS 的设定，要比采集 byteCount 个数据所需时间长，否则当函数返回时有可能无法返回 byteCount 个数据，此时需要使用通过 realByteCount 来判断返回了多少个数，会增加编程工作量。

3.设置合理的情况下，byteCount 和 realByteCount 应该相等。

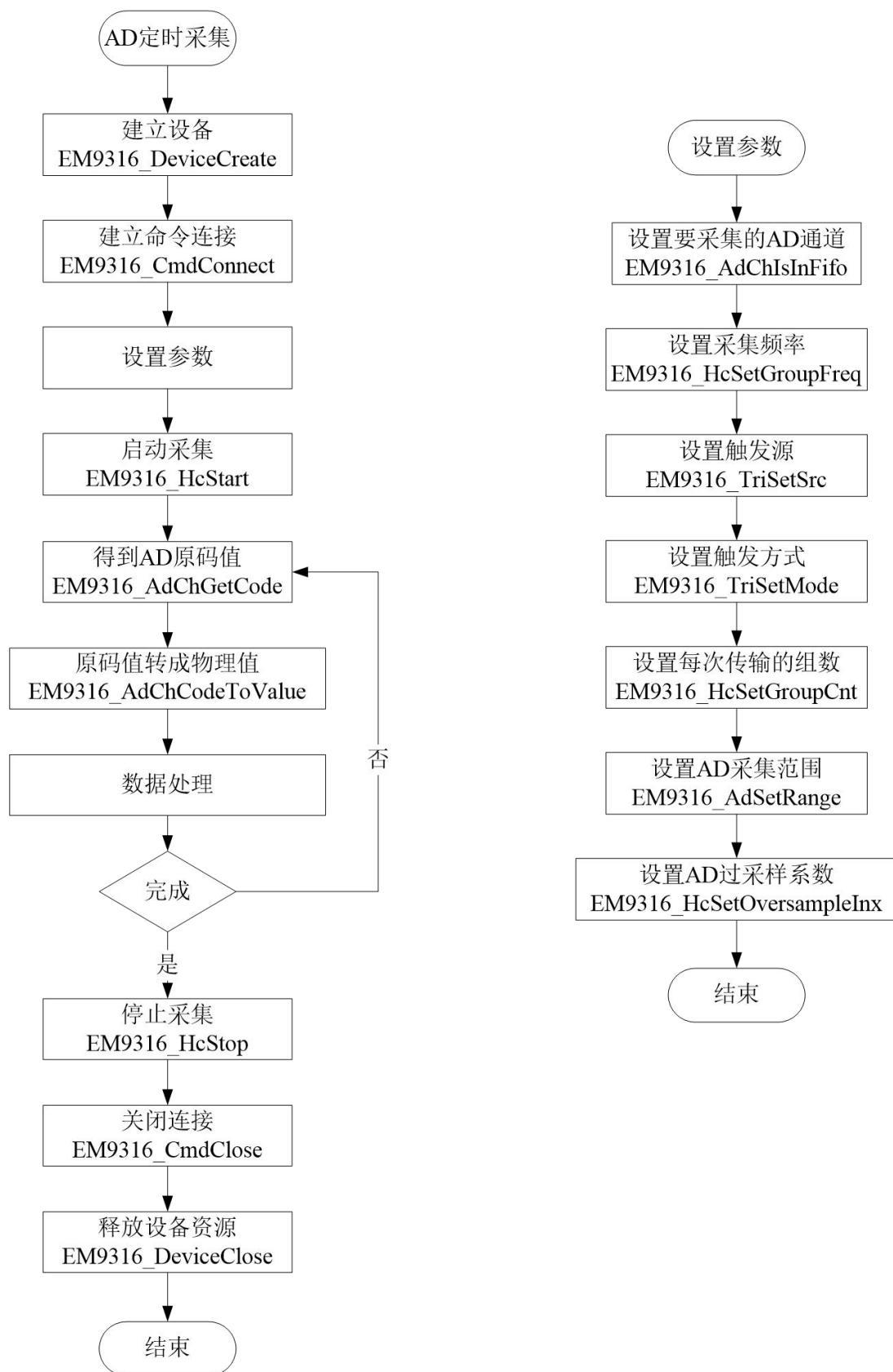
4.调用完此函数后，可以使用 EM9316_AdChGetCode 和 EM9316_AdChCodeToValue 两个函数将原码值转换成物理值。

5.可以调用 EM9316_SFifoCanReadBC 读取上位机缓冲区数据量，EM9316_HcFifoStatus 读取下位机缓冲区数据量。如果上位机缓冲区数据量一直处于增长状态，说明程序读取数据不及时。如果下位机缓冲区数据量一直处于增长状态，说明网络状态不好，通讯速度慢。

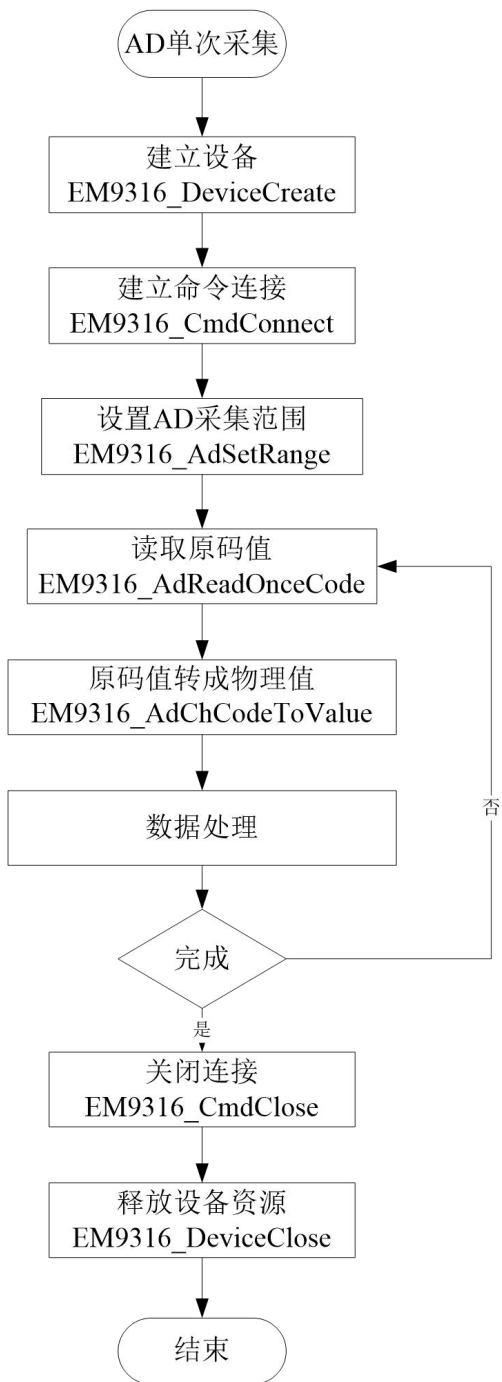
6.可以调用 EM9316_AdGetMaxChCnt 读取下位机 AD 通道数，EM9316B-8 最大 8 通道，

EM9316B-16 最大 16 通道，使能 AD 通道数大于最大通道数没有意义。

流程图如下：



2.2.3 单次采集（慢速采集）：



2.3 开关量编程方法

2.3.1 相关函数：

EM9316_IoSetDir

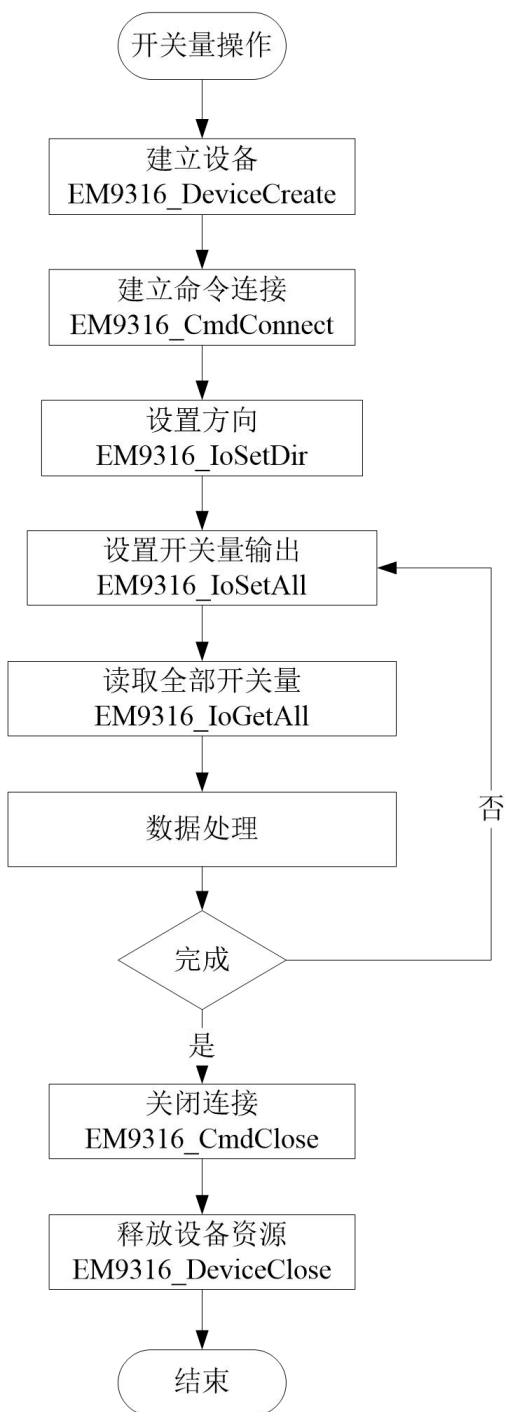
EM9316_IoSetAll

EM9316_IoGetAll

2.3.2 IO、DO 与 DI 操作：

本设备具有 16 路 IO，8 路一组设置方向，可使用 EM9316_IoSetDir 设置成输出或者输入。8 路 DO 固定输出，8 路 DI 固定输入。EM9316_IoSetAll 函数设置全部开关量端口，如果设置的端口为输入，则忽略。EM9316_IoGetAll 读取全部开关量端口状态，注意本卡 DO 是反逻辑，输出 1 光耦关断，输出 0 光耦导通。设置和读取函数都传递了 32 元素数组，元素 0~15 对应 IO1~16，元素 16~23 对应 DI1~8，元素 24~31 对应 DO1~8。

操作流程图：



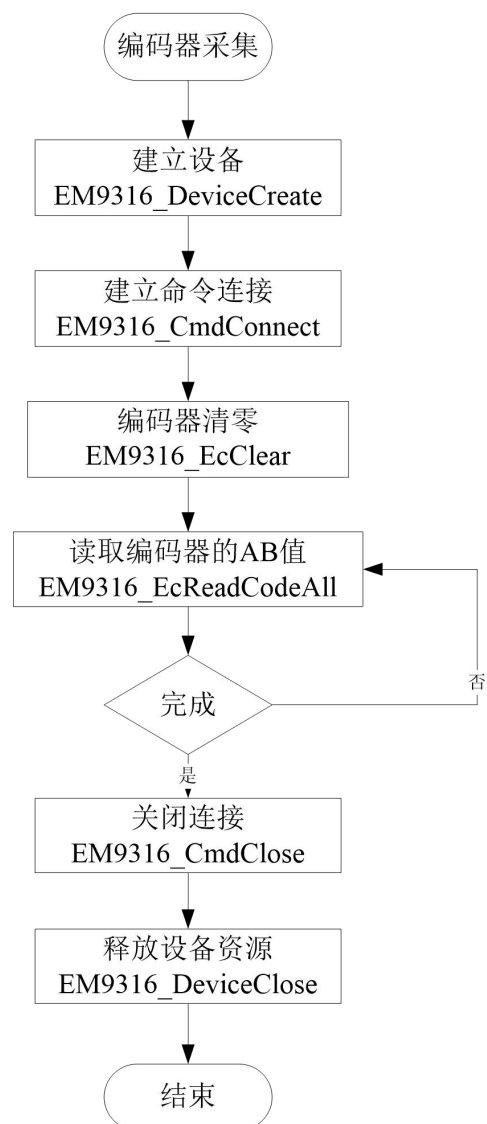
2.4 编码器编程方法

2.4.1 相关函数：

EM9316_EcClear
EM9316_EcReadCodeAll

2.4.2 编码器采集：

EM9316B 具有 2 路编码器接口，可以测量正交编码器，与计数器接口共用。
流程图如下：



2.5 计数器编程方法

本设备具有 6 路计数器输入，能够测频、计数，还可以设置滤波系数来增加抗干扰性。

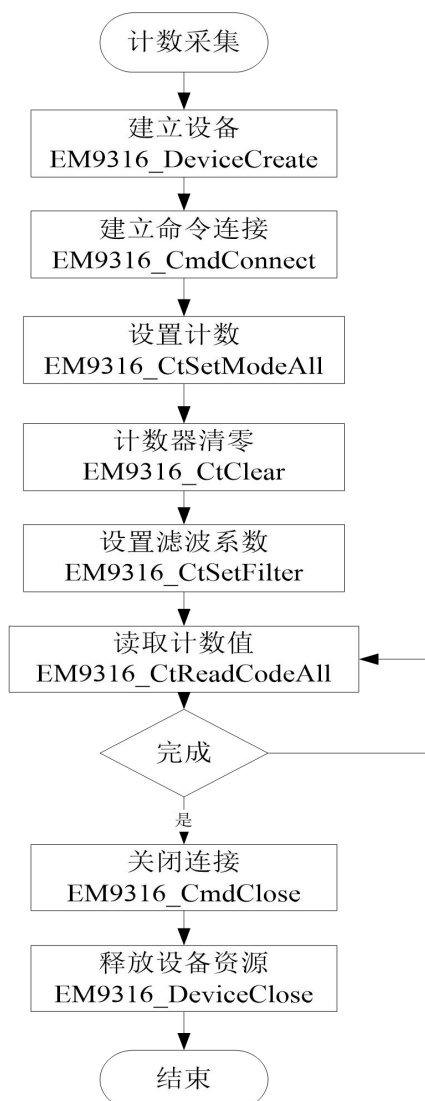
2.5.1 相关函数：

EM9316_CtSetModeAll
EM9316_CtClear
EM9316_CtSetFreqBase
EM9316_CtReadCodeAll
EM9316_CtCodeToValue
EM9316_CtSetFilter

2.5.2 计数采集：

在计数过程中，用户可以随时使用 EM9316_CtClear 将计数值清零。

流程图如下：



2.5.3 频率采集:

本设备的测频基准时钟是由 16MHz(16,000,000Hz)时钟分频而得,因此测频基准最小值是 62.5nS(0.0000625mS)。最大值是 $1000/16*(2^{32})nS$ (268,435,456uS, 约为 268 秒),用到最大值的可能性几乎不存在。

可以根据被测信号频率的快慢选择测低频还是测高频,采集精度可以通过下面的公式得到:

测高频方式

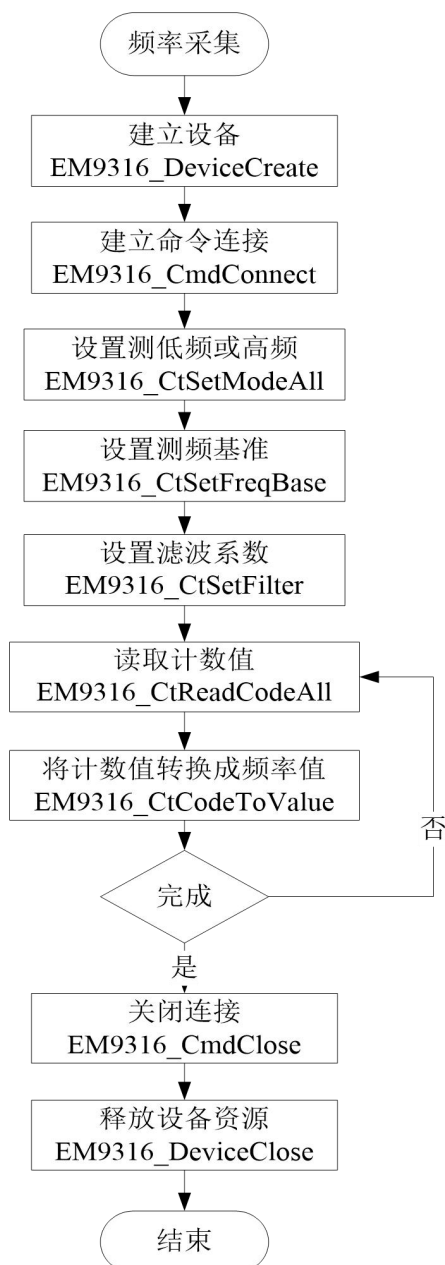
采集精度 = 被测信号周期 / 基准时钟周期 * 100%

测低频方式

采集精度 = 基准时钟周期 / 被测信号周期 * 100%

用户要根据自己的被测信号的特点选择合适的工作方式,这样才能够获得较高的测频精度与测频速度。若用户不清楚被测信号的频率范围,可以先使用测低频的方式获得被测信号的大致频率值。如果发现被测频率较高,则可以进一步使用测高频工作方式得到较为精确的频率值。

流程图如下:



2.6 PWM 编程方法

2.6.1 相关函数：

EM9316_PwmStartAll
EM9316_PwmSetPulse
EM9316_PwmIsOver
EM9316_PwmSetCount
EM9316_PwmSetBaseClk

2.6.2 PWM 操作：

本设备的 PWM 输出与数字量输出管脚复用，只有在 PWM 输出处于非使能状态下时才能操作相应的数字量。

本设备的 PWM 脉冲输出不但支持**频率**和**占空比**的调整，还可以进行**脉冲个数**与**相位**的设置。

PWM 脉冲输出使用 16MHz (16,000,000Hz)分频后的时钟作为基准频率，通过设置**分频系数**来改变**输出频率**，参见下面公式：

基准时钟频率 = 16000000Hz / (基准时钟分频系数 + 1)

输出频率 = 基准时钟频率 / 分频系数

占空比 = 高电平系数 / 分频系数

所有系数都是 16 位。

注意：当所设定的**输出频率**越高时，**占空比**的精度越低，详见 4.3 节。

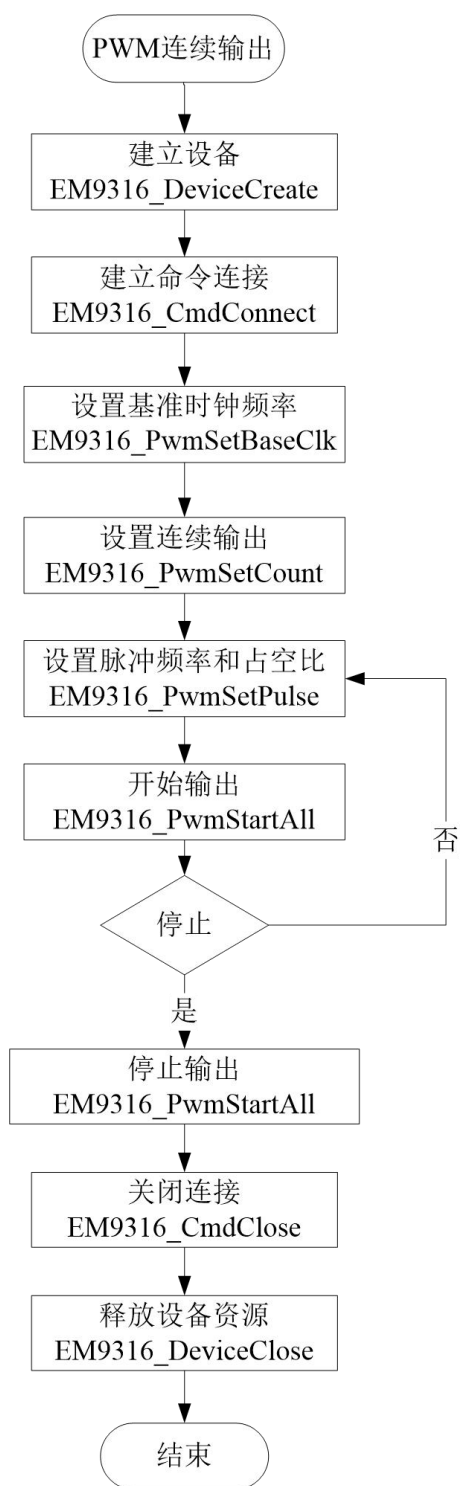
脉冲个数的取值范围是 0~65535，当其设置为 0 时，表示连续输出。

相位的设置支持两种状态：不延迟和延迟 90° 输出。这样可以使用两路 PWM 输出控制一些步进电机驱动器，和本设备的编码器输入功能一起形成伺服控制系统。

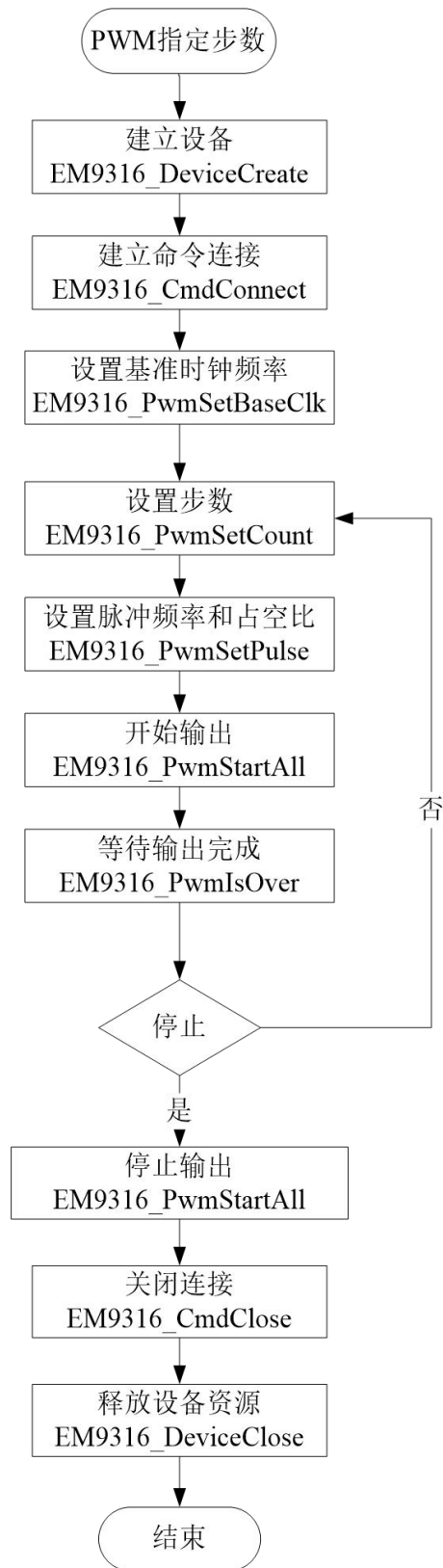
要进一步了解 PWM 操作原理请看“[4.3 PWM 频率输出原理](#)”

具体编程请参考后面连续 PWM 脉冲输出和指定步数 PWM 脉冲输出的流程图。

连续 PWM 脉冲输出流程图：



指定步数 PWM 脉冲输出流程图：



2.7 DA 编程方法

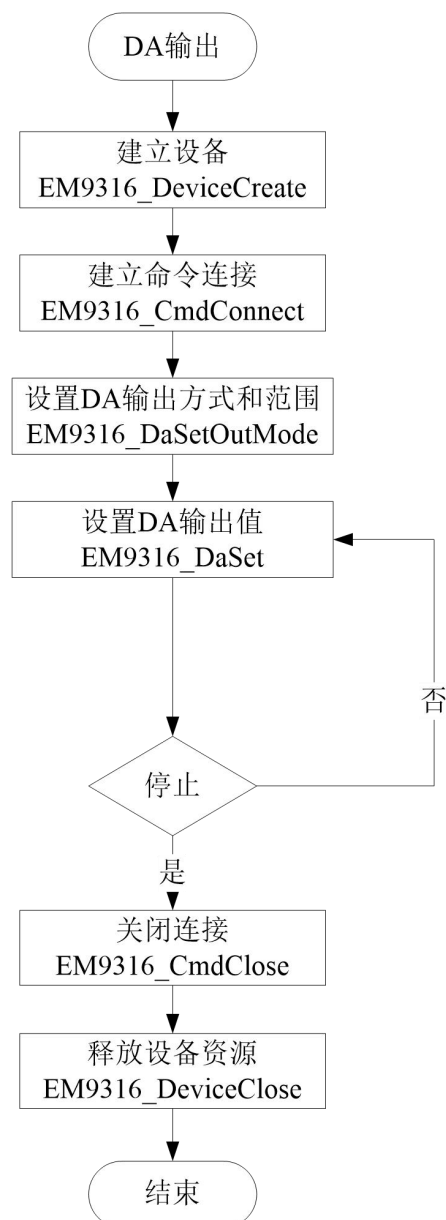
2.7.1 相关函数：

EM9316_PwmStartAll
EM9316_PwmSetPulse
EM9316_PwmIsOver
EM9316_PwmSetCount

2.7.2 DA 操作：

本设备是 16 位 DA，出厂已经校准，可以直接使用电压或者电流值方式输出，具体要和下位机跳线对应，也可以使用原码值输出，此时需要用户自行校准。

具体操作流程如下：



2.8 高速同步采集编程方法

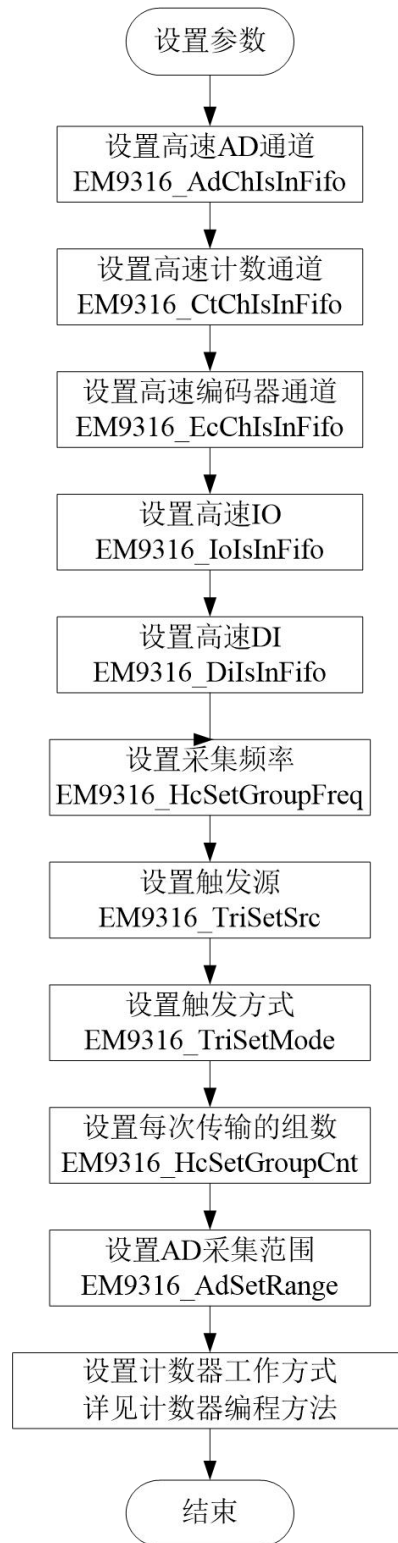
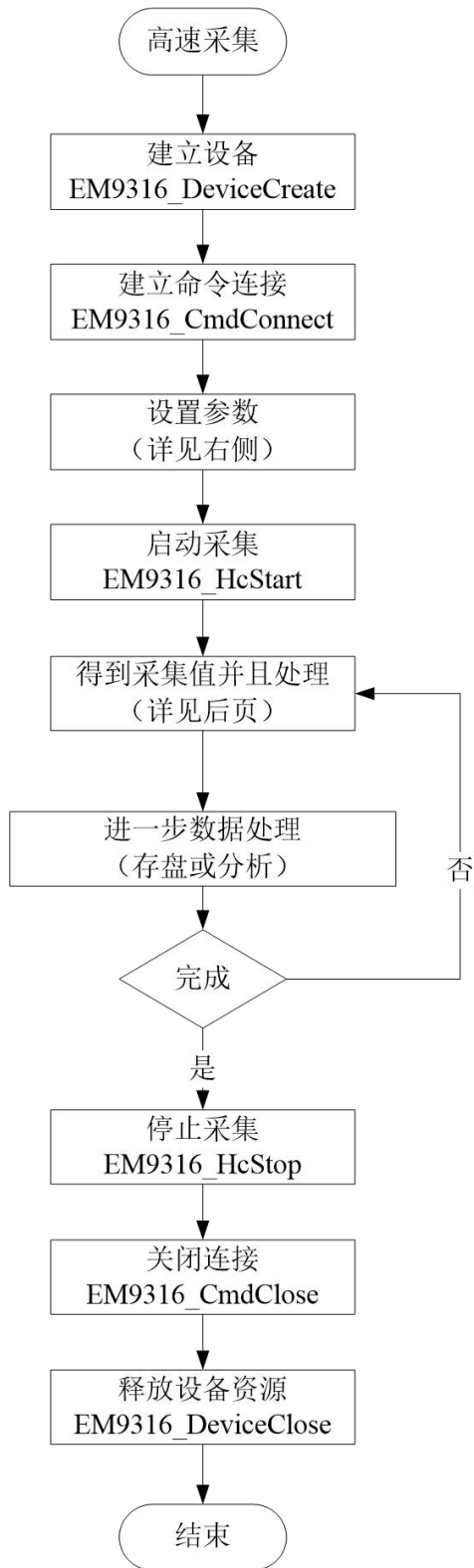
2.8.1 相关函数：

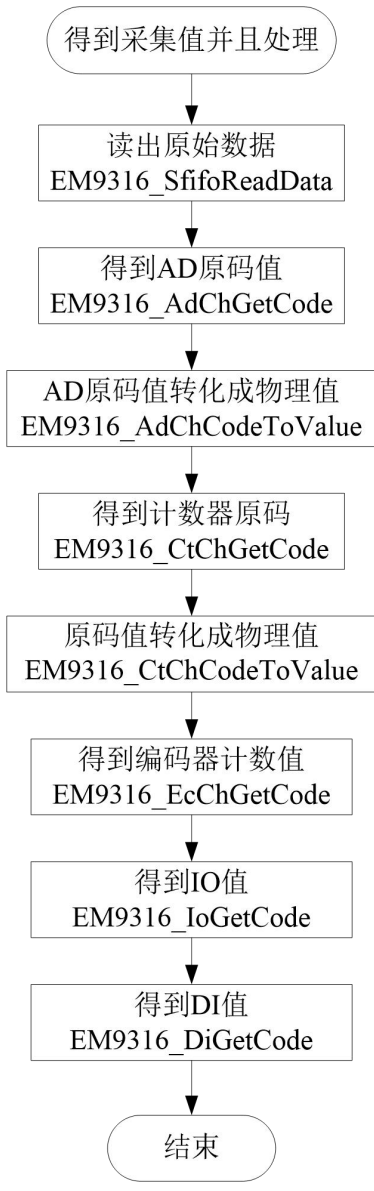
EM9316_AdChIsInFifo
EM9316_CtChIsInFifo
EM9316_EcChIsInFifo
EM9316_IoIsInFifo
EM9316_DiIsInFifo
EM9316_SfifoReadData
EM9316_AdChGetCode
EM9316_CtChGetCode
EM9316_CtChCodeToValue
EM9316_EcChGetCode
EM9316_IoGetCode
EM9316_DiGetCode

2.8.2 高速采集：

用户通过相关函数可以设定对应的通道是否进入缓冲区（Fifo），进入缓冲区的数据为硬件采集时钟控制启动采集结果，每一个采集时钟上升沿触发一次采集，每次采集包括所有设定进入缓冲区的通道的数据。

流程图见后页：





2.9 读取离线文件编程方法

2.9.1 相关函数：

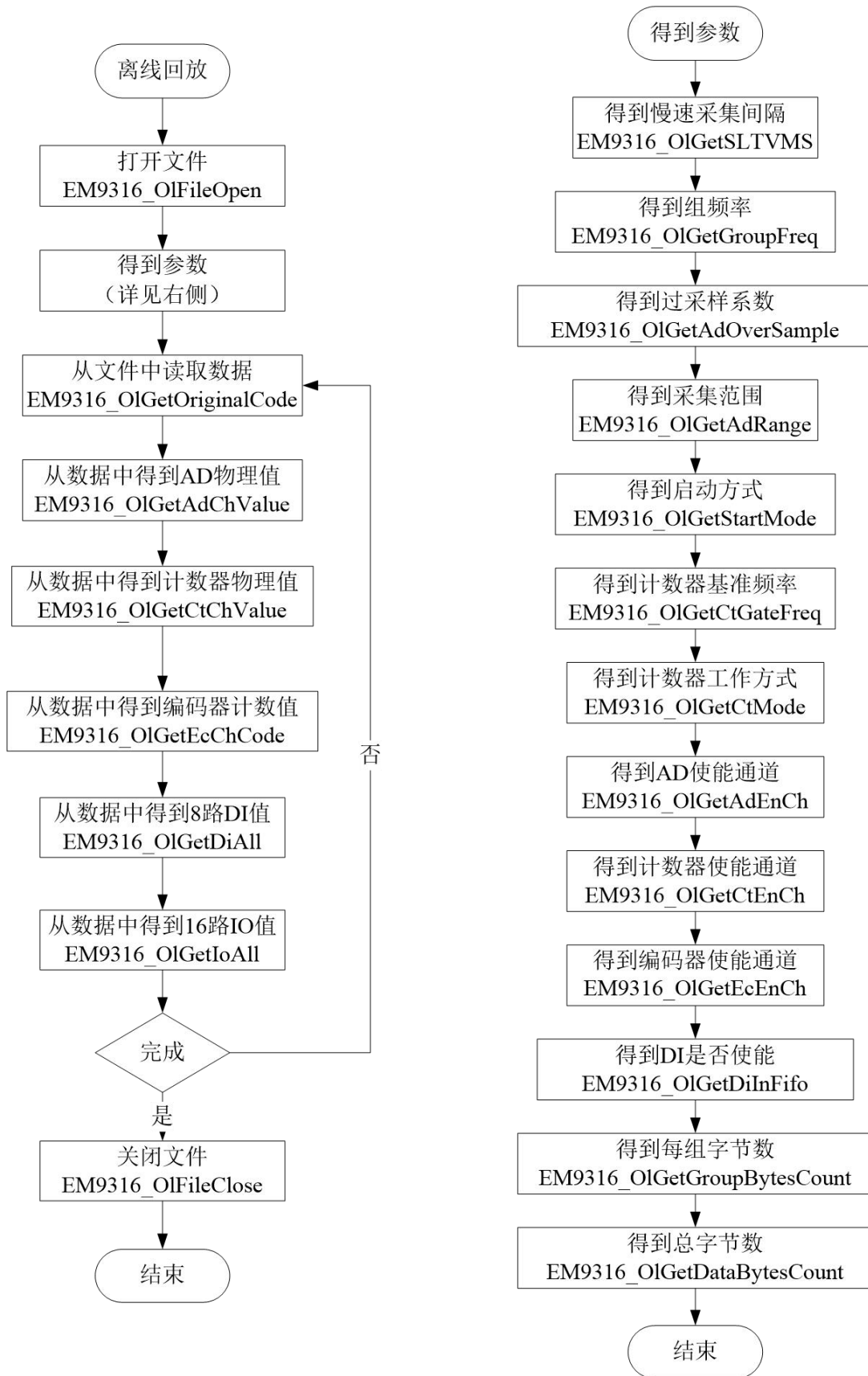
EM9316_OIFileOpen
EM9316_OIGetSLTVMS
EM9316_OIGetGroupFreq
EM9316_OIGetAdRange
EM9316_OIGetStartMode
EM9316_OIGetCtGateFreq
EM9316_OIGetCtMode
EM9316_OIGetAdEnCh
EM9316_OIGetCtEnCh
EM9316_OIGetEcEnCh
EM9316_OIGetDiInFifo
EM9316_OIGetGroupBytesCount
EM9316_OIGetDataBytesCount
EM9316_OIGetAdOverSample
EM9316_OIGetOriginalCode
EM9316_OIGetAdChValue
EM9316_OIGetCtChValue
EM9316_OIGetEcChCode
EM9316_OIGetDiAll
EM9316_OIGetIoAll

2.9.2 读取离线文件：

离线数据文件是指本设备在离线采集时存储的文件，它是二进制格式，文件分为文件头和数据区。文件头储存了一些必要的参数设置；数据区将采集到的数据以原码值的方式将全部使能通道数据放到一起。

要处理离线数据文件，首先要使用相关函数读取出设备在采集存储时的使能通道和采集频率，然后读取出数据原码。当数据文件比较大时，建议分批读取数据，否则可能会发生内存溢出等错误。每次读取的数据量最好是每组数据数目的整数倍，这样可以减少工作量。获取数据原码后，通过几个相关函数可以提取出相应的 AD、计数器以及编码器的采集结果。用户可以进一步对采集结果进行处理。

具体操作请参考后页流程图：



第三章 函数原型与功能详解

本章函数原型均为 C 语言方式，VC++可以在包含了“ZT_Type.h”头文件后直接使用下面的声明。

3.1 数据类型

I8: 8 位有符号整型

U8: 8 位无符号整型

I16: 16 位有符号整型

U16: 16 位无符号整型

I32: 32 位有符号整型

U32: 32 位无符号整型

F64: 64 位双精度浮点型

3.2 函数说明

请见“EM9316.h”文件

第四章 基本原理

4.1 物理值转换与数据格式

本节主要介绍数据采集设备如何在缓冲区中存放数据，以及如何将这些数据转换成有意义的物理量。

4.1.1 物理值转换

EM9316 具有模拟量、计数器、编码器和开关量输入通道，每种输入通道的数据格式和转换方式均有所不同，下面分别介绍。

4.1.1.1 模拟量输入：16 位有符号整型

原码到物理值的转换：

原码还需要经过相应的转换才能得到实际的电压值或者电流值，我们称转换后的值为**物理值**。

本数据采集设备原码是 16 位二进制补码。理想情况下，采集范围内最小输入物理值对应的原码值是 -32768，最大输入物理值对应的原码值是 32767，零点值对应的是 0。但是在实际中，每个通道会有点细微的差别。为了补偿这个差别，我们在生产过程中使用标准信号对设备进行校准，将每一路的 AD 信号对应的实际零点值和满度值记录到设备的 ROM 中。用户使用“读取零点满度信息”命令可以得到所有通道的零点值和满度值。得到这两个值后，可以计算出实际的物理值。

我们储存的零点值是在接地（0V）时对应的原码值，满度值是满度电压对应的原码值，原码值与

物理值的换算关系如下：

±10V:

$$\text{实际电压值} = (\text{原码值} - \text{零点值}) / (\text{满度值} - \text{零点值}) * 9(V)$$

±5V:

$$\text{实际电压值} = (\text{原码值} - \text{零点值}) / (\text{满度值} - \text{零点值}) * 4.5(V)$$

4.1.1.2 计数器输入：16 位无符号整型

从下位机读回的数据是原码值

计数器工作在计数模式下时，其原码值直接就对应脉冲个数，代表了自从上次清零以来相应通道接受到的脉冲个数。

计数器工作在测频模式下时，需要进行转换才能得到实际频率值，具体转换方式请参考后面的“计数器测频原理”

4.1.1.3 编码器输入：32 位有符号整型

本设备只能测量正交编码器，从下位机读回的数据是原码值，这个值是端口脉冲的四倍频值。

4.1.1.4 IO 输入：16 位无符号整型

Bit0~15 对应 IO1~16，当对应通道设成输出时，返回的就是之前设置的输出状态。

4.1.1.5 DI 输入：8 位无符号整型

Bit0~8 对应 DI1~8

4.1.2 高速缓冲区数据格式

本数据采集设备的高速缓冲区是一个先进先出缓冲区，简称为 FIFO。FIFO 的特点是最早采集到的数据将会首先被用户读取到。

只有允许进入 FIFO 的数据才会包含在高速缓冲区中，在允许全部输入通道进入 FIFO 的情况下，一组数据的长度将达到 56 字节，每个字节含义如下表所示

字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6	字节 7
AD1 Bit0~Bit7	AD1 Bit8~Bit15	AD2 Bit0~Bit7	AD2 Bit8~Bit15	AD3 Bit0~Bit7	AD3 Bit8~Bit15	AD4 Bit0~Bit7	AD4 Bit8~Bit15
字节 8	字节 9	字节 10	字节 11	字节 12	字节 13	字节 14	字节 15
AD5 Bit0~Bit7	AD5 Bit8~Bit15	AD6 Bit0~Bit7	AD6 Bit8~Bit15	AD7 Bit0~Bit7	AD7 Bit8~Bit15	AD8 Bit0~Bit7	AD8 Bit8~Bit15
字节 16	字节 17	字节 18	字节 19	字节 20	字节 21	字节 22	字节 23
AD9 Bit0~Bit7	AD9 Bit8~Bit15	AD10 Bit0~Bit7	AD10 Bit8~Bit15	AD11 Bit0~Bit7	AD11 Bit8~Bit15	AD12 Bit0~Bit7	AD12 Bit8~Bit15

字节 24	字节 25	字节 26	字节 27	字节 28	字节 29	字节 30	字节 31
AD13 Bit0~Bit7	AD13 Bit8~Bit15	AD14 Bit0~Bit7	AD14 Bit8~Bit15	AD15 Bit0~Bit7	AD15 Bit8~Bit15	AD16 Bit0~Bit7	AD16 Bit8~Bit15
字节 32	字节 33	字节 34	字节 35	字节 36	字节 37	字节 38	字节 39
CT1 Bit0~Bit7	CT1 Bit8~Bit15	CT2 Bit0~Bit7	CT2 Bit8~Bit15	CT3 Bit0~Bit7	CT3 Bit8~Bit15	CT4 Bit0~Bit7	CT4 Bit8~Bit15
字节 40	字节 41	字节 42	字节 43	字节 44	字 45	字节 46	字节 47
CT5 Bit0~Bit7	CT5 Bit8~Bit15	CT6 Bit0~Bit7	CT6 Bit8~Bit15	EC1 Bit0~Bit7	EC1 Bit8~Bit15	EC1 Bit16~Bit23	EC1 Bit24~Bit31
字节 48	字节 49	字节 50	字节 51	字节 52	字 53	字节 54	字节 55
EC2 Bit0~Bit7	EC2 Bit8~Bit15	EC2 Bit16~Bit23	EC2 Bit24~Bit31	IO1~IO8 Bit0~Bit7	IO9~IO16 Bit0~Bit7	DI1~DI8 Bit0~Bit7	DO1~DO8 Bit0~Bit7
.....							
循环							

在上表中，AD 代表模拟量输入，CT 代表计数器输入，EC 代表编码器输入。这些数据均为小端排列，也就是对于每个通道，排列在前的字节是数据的低位。

每组数据均是同一时刻采集到的数据，我们建议每次处理数据时，数据量为每组字节数整数倍，这样无需考虑通道数据对齐问题。进入 FIFO 的通道不同，每组字节数也会有所不同，需要根据实际情况计算。

下面再举几个物理量进入 FIFO 的例子。

当仅允许所有 16 路 AD 通道进入 FIFO 时，每组字节数为 32，含义如下表：

字节序号	0~1	2~3	4~5	6~7	8~9	10~11	12~13	14~15
含义	AD1	AD2	AD3	AD4	AD5	AD6	AD7	AD8
字节序号	16~17	18~19	20~21	22~23	24~25	26~27	28~29	30~31
含义	AD9	AD10	AD11	AD12	AD13	AD14	AD15	AD16

当只有 AD0~AD5 允许进入 FIFO 时，每组字节数为 12，含义如下表：

字节序号	0~1	2~3	4~5	6~7	8~9	10~11
含义	AD1	AD2	AD3	AD4	AD5	AD6

当只有计数器允许进入 FIFO 时，每组字节数为 12，含义如下表：

字节序号	0~1	2~3	4~5	6~7	8~9	10~11
含义	CT1	CT2	CT3	CT4	CT5	CT6

4.2 计数器测频原理

当用户将计数器设置成测频方式时，需要设置**测频基准时间**，计数器内部将会产生以**测频基准时间**为周期的脉冲，称为**基准脉冲**。**测频基准时间**是通过设置**计数器测频基准分频系数**分频**内部基准时钟**来获得的。

测频基准时间 = 计数器测频基准分频系数 / 内部基准时钟频率

本设备的内部基准时钟频率是 16MHz (16,000,000Hz)。因此测频基准时间最小值约为 62.5 纳秒 (0.0000625 毫秒)，由此决定设置的时间只能是 62.5 纳秒的整数倍。

测频精度的基本公式如下：

$$\text{测频精度} = 1 / \text{计数器值}$$

在不同测频方式下计数器值代表的含义不同。

测高频时（一般 1KHz 以上），此时计数器值是在一个基准脉冲周期内被测信号的脉冲个数。

$$\text{计数器值} = \text{被测信号频率 (赫兹)} * \text{测频基准时间 (秒)}$$

测高频精度计算举例：当测频基准时间设置为 1 秒 (1000 毫秒) 时，测量 10KHz (10000Hz) 的信号，每秒钟计数值为 10000，其精度为 $1/10000=0.0001=0.01\%$ ；如果将测频基准时间改为 100 毫秒，则 100 毫秒内的计数值是 1000，其精度为 $1/1000=0.001=0.1\%$ ；

测低频时（一般 1KHz 以下），此时计数器值是被测信号一个周期内基准脉冲的个数。

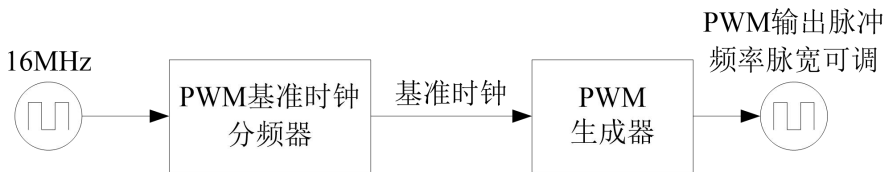
$$\text{计数器值} = 1 / \text{低频被测信号频率 (赫兹)} * \text{测频基准时间 (秒)}$$

测低频精度计算举例：当测频基准时间设置为 10^{-6} 秒 (0.001 毫秒) 时，测量 100Hz 的信号，则计数器值为 10000，其精度为 $1/10000=0.0001=0.01\%$ 。

用户要根据自己被测信号的特点选择合适的工作方式，这样才能够使用 EM108C 获得较高的测频精度与测频速度，如果用户不清楚被测信号的频率范围，可以先使用测低频的方式获得被测信号的大致频率值，如果发现被测频率较高，则可以进一步使用测高频工作方式得到较为精确的频率值。

4.3 PWM 频率输出原理

PWM 输出原理图：



本设备使用 16MHz 时钟作为 PWM 输出时钟，经过 PWM 基准时钟分频器生成 PWM 基准时钟给 PWM 生成器，最终生成脉宽、频率和步数均可调节的 PWM 脉冲。本设备 PWM 操作相关寄存器都是 16 位，因此最大值是 65535。下面三个公式可计算出 PWM 脉冲输出频率和占空比。

$$\text{PWM 基准时钟频率} = 16000000 / (\text{基准时钟分频系数} + 1)$$

$$\text{PWM 脉冲频率} = \text{基准时钟频率} / \text{PWM 分频系数}$$

$$\text{PWM 占空比} = \text{高电平系数} / \text{PWM 分频系数}$$

由上面三个公式可以看出，最终的 PWM 输出频率和占空比会有数字化误差，误差大小和分频系数成反比，也就是分频系数越大，误差越小。

当基准时钟频率为 16MHz 时，能生成的最小频率= $16000000/65535 \approx 244.14\text{Hz}$ ，生成 1000Hz 需要的 PWM 分频系数是 16000，占空比调节最小变化差是 $1/16000$ 。1000Hz 下一个频率分频系数是 16001，则输出频率= $1600000/16001 \approx 999.94\text{Hz}$ ；

当基准时钟频率为 160KHz 时，能生成的最小频率=160000/65535 \approx 2.44Hz，生成 1000Hz 需要的 PWM 分频系数是 160，占空比调节最小变化差是 1/160。1000Hz 下一个频率分频系数是 161，则输出频率=160000/161 \approx 993.79Hz；

从上面两个例子可以看出，基准时钟频率越小，能生成的最小频率越小，但是在输出较高频率时，占空比调节和频率调节挡位就会越大，因此如果想要获得较高的输出精度，需要在满足输出频率的基础上将基准时钟频率设置的尽可能大。

更新记录

时间	更改内容
2024-1-23	初次发布
2024-2-4	修正 PWM 频率输出原理
2024-2-5	修正部分字符错误
2024-4-29	流程图中增加 EM9316_PwmSetBaseClk 函数