EM9118 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

目 录

第一章 使用动态链接库编程方法

1.1 概述

EM9118 是 16 位 450KHz 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。

在使用本函数库时，请确保“ZTLC_TCP.dll”与“EM9118.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。

1.2 模拟量（AD）编程方法

相关函数：

EM9118_DeviceCreate

EM9118_DeviceClose

EM9118_CmdConnect

EM9118_DataConnect

EM9118_CmdClose

EM9118_DataClose

EM9118_AdChIsInFifo

EM9118_AdSetRange

EM9118_HcSetGroupFreq

EM9118_HcStart

EM9118_HcStop

EM9118_HcReadData

EM9118_GetFifoGroupByteCount

EM9118_AdChGetCode

EM9118_AdChBatchCodeToValue

内部定时采集：

本设备板载 64MB 缓存，可以调用 EM9118_HcReadData 函数读回缓冲区中的数据。缓冲区是 FIFO 形式，数据读出后就不再保存在缓冲区中。

关于 EM9118_HcReadData 函数的使用，有下面几个建议：

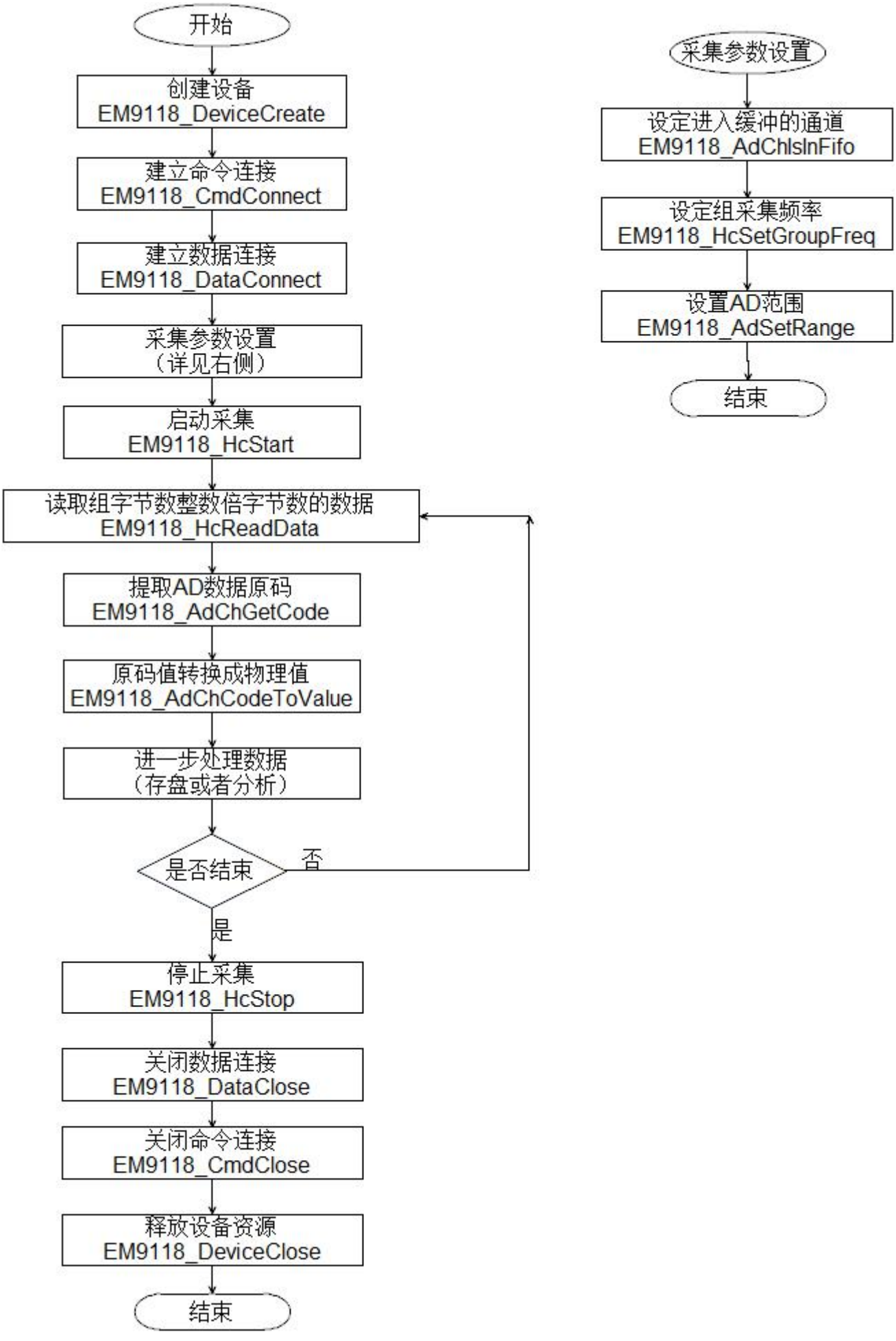
1.每次读出的数据字节数（byteCount）为缓冲区中每组字节数的整数倍，这样便于后续的数据处理。可以通过 EM9118_GetFifoGroupByteCount 函数得到每组字节数。

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，此时会增加编程工作量。

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 EM9118_AdChGetCode 和 EM9118_AdChCodeToValue 两个函数将原码值转换成物理值。

流程图见后页：


开始
创建设备
EM9118_DeviceCreate
建立命令连接
EM9118_CmdConnect
建立数据连接
EM9118_DataConnect
采集参数设置
（详见右侧）
启动采集
EM9118_HcStart
读取组字节数整数倍字节数的数据
EM9118_HcReadData
提取AD数据原码
EM9118_AdChGetCode
原码值转换成物理值
EM9118_AdChCodeToValue
进一步处理数据
（存盘或者分析）
是否结束
否
是
停止采集
EM9118_HcStop
关闭数据连接
EM9118_DataClose
关闭命令连接
EM9118_CmdClose
释放设备资源
EM9118_DeviceClose
结束
采集参数设置
设定进入缓冲的通道
EM9118_AdChIsInFifo
设定组采集频率
EM9118_HcSetGroupFreq
设置AD范围
EM9118_AdSetRange
结束

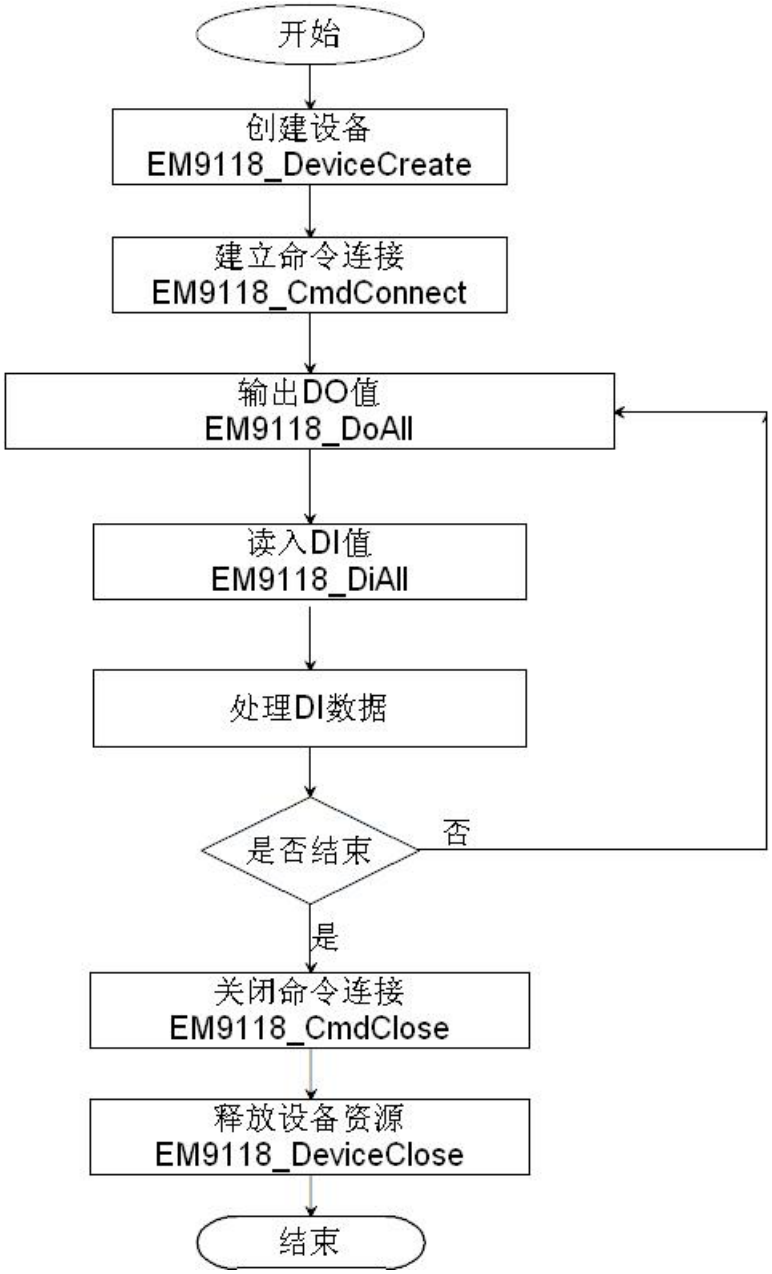
1.3 开关量编程方法

1.3.1 相关函数：

EM9118_DiAll
EM9118_DoAll

1.3.2 IO 操作：

IO 读写流程图：



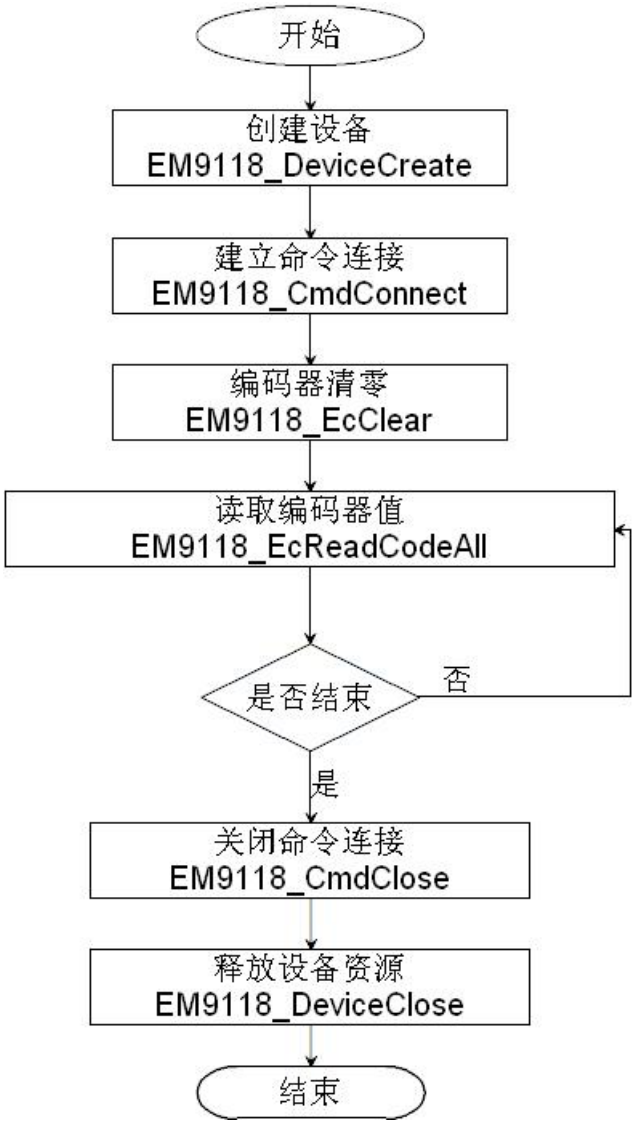
1.4 编码器编程方法

1.4.1 相关函数：

EM9118_EcClear
EM9118_EcReadCodeAll

1.4.2 编码器采集：

流程图如下：



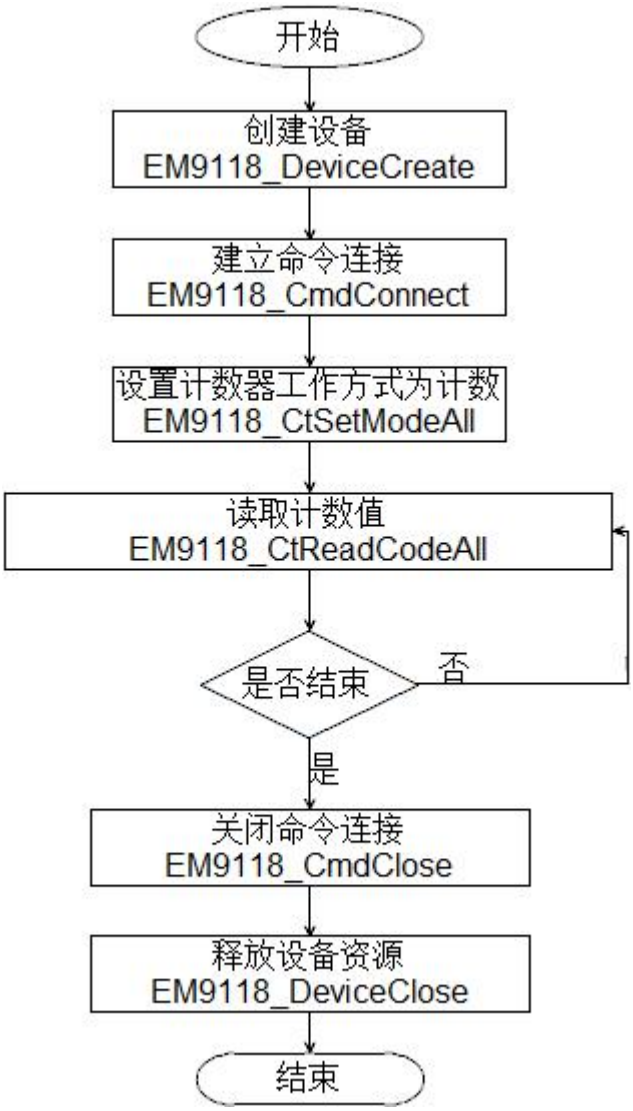
1.5 计数器编程方法

1.5.1 相关函数：

EM9118_CtSetModeAll
EM9118_CtClear
EM9118_CtSetFreqBase
EM9118_CtReadCodeAll
EM9118_CtChBatchCodeToValue

1.5.2 计数采集：

在计数过程中，用户可以随时使用 EM9118_CtClear 将计数值清零。
流程图如下：



1.5.3 频率采集：

本设备的测频基准时钟是由 36MHz（36,000,000Hz）时钟分频而得，因此测频基准最小值是 27.78nS（0.0000278mS）。最大值是 1000/36*(2^32)nS（119,304,647uS，约为 119 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

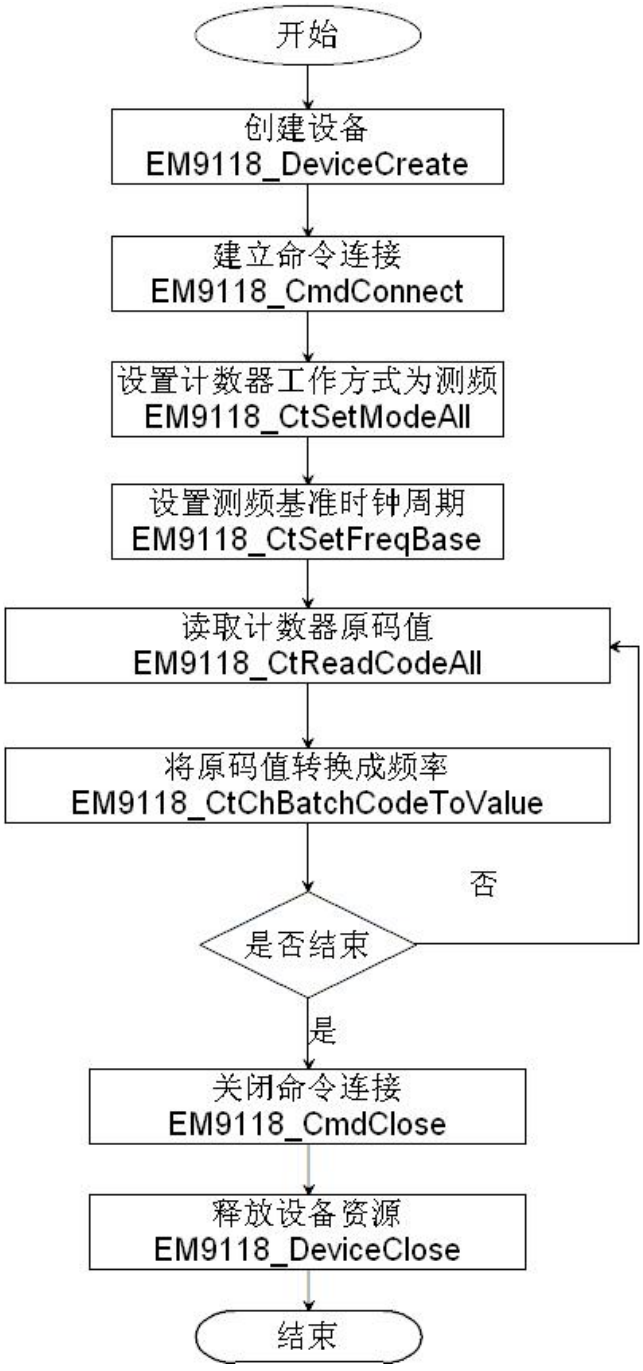
采集精度 = 被测信号周期 / 基准时钟周期 * 100%

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：



1.6 PWM 编程方法

1.6.1 相关函数：

EM9118_PwmStartAll
EM9118_PwmSetPulse
EM9118_PwmIsOver
EM9118_PwmSetCount

1.6.2 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 4MHz（4,000,000Hz)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：

输出频率 = 4,000,000Hz / 分频系数

分频系数是 16 位，输出频率范围约为 70Hz~1MHz

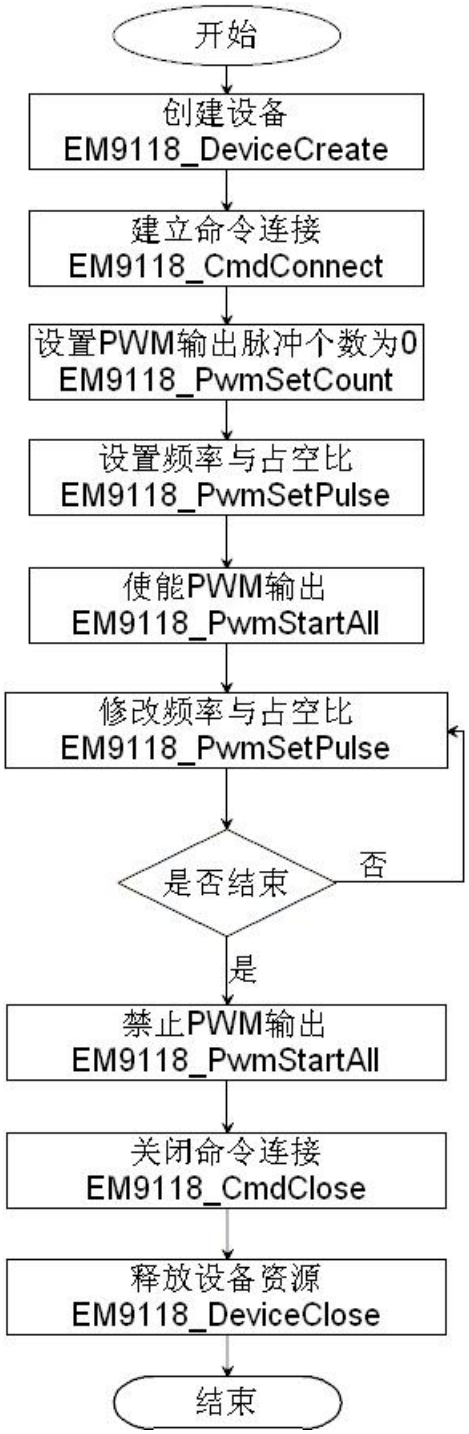
注意：当所设定的**输出频率**越高时，**占空比**的精度越低。

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。

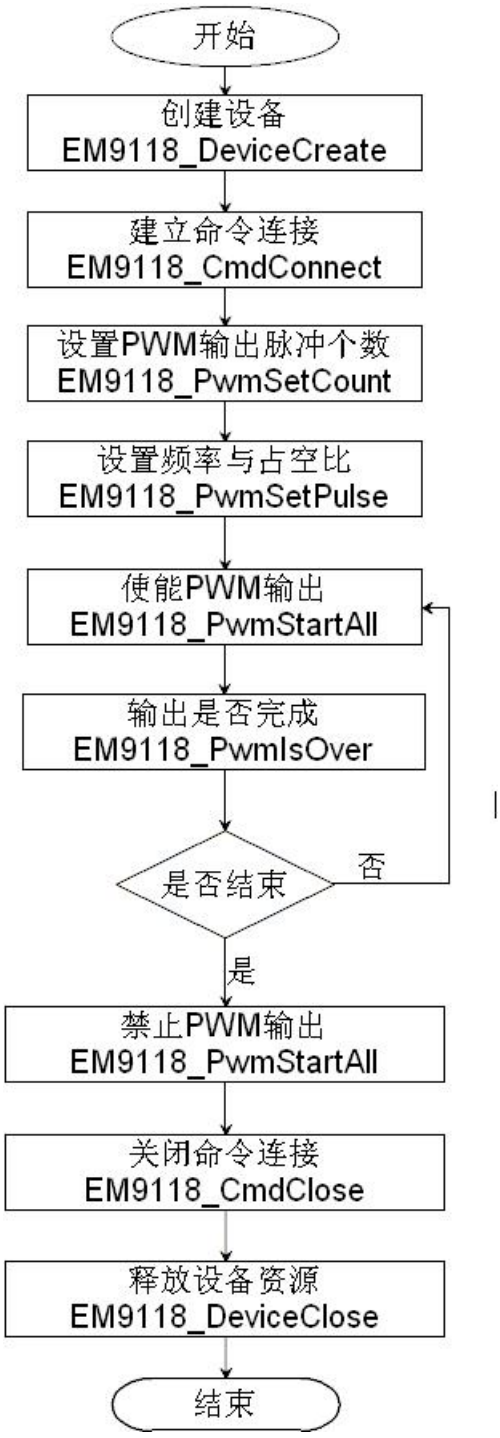
相位的设置支持两种状态：不延迟和延迟 90° 输出。这样可以使用两路 PWM 输出控制一些步进电机驱动器，和本设备的编码器输入功能一起形成伺服控制系统。

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：



1.7 高速同步采集编程方法

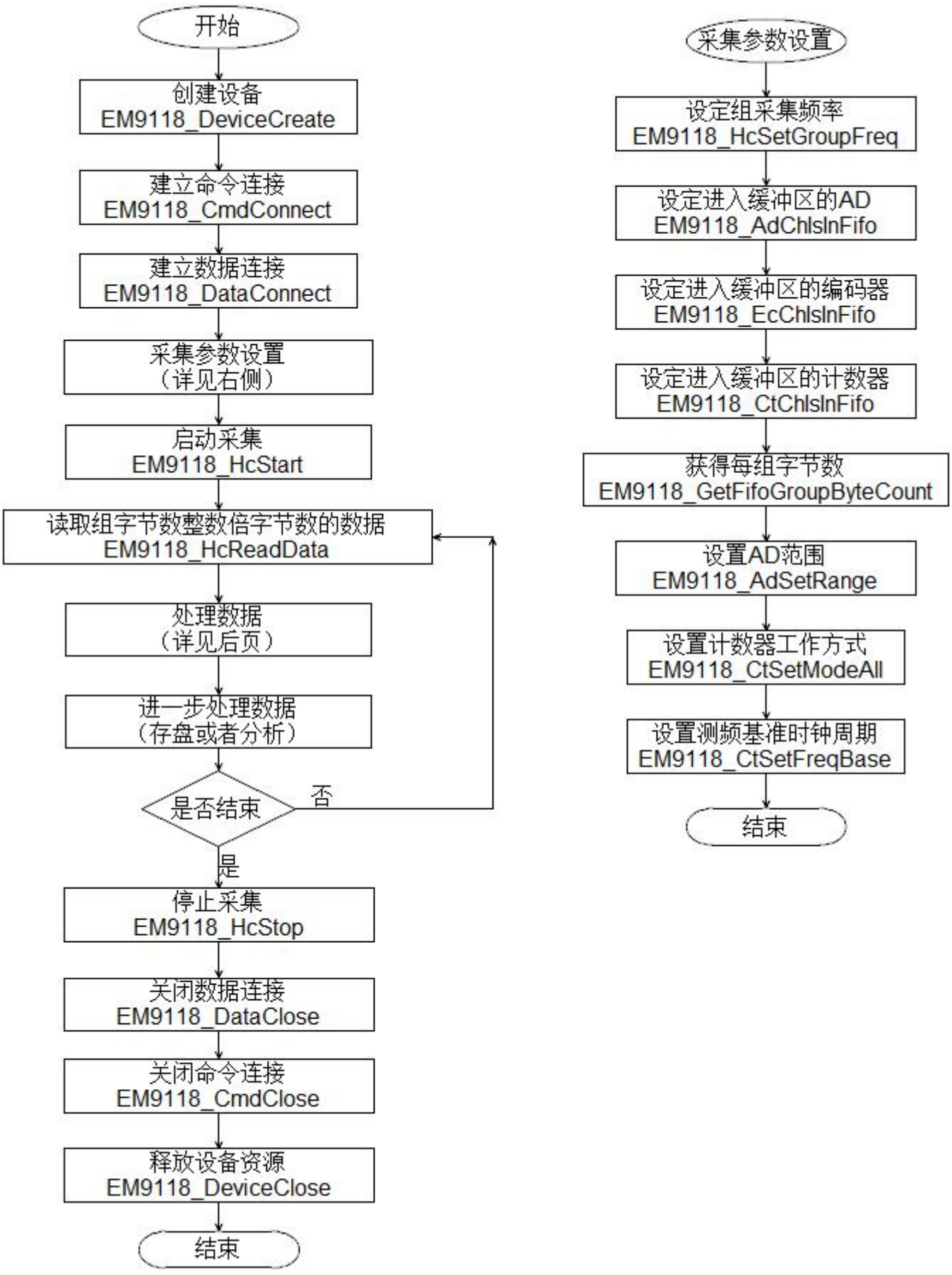
1.7.1 相关函数（仅列出新增函数）：

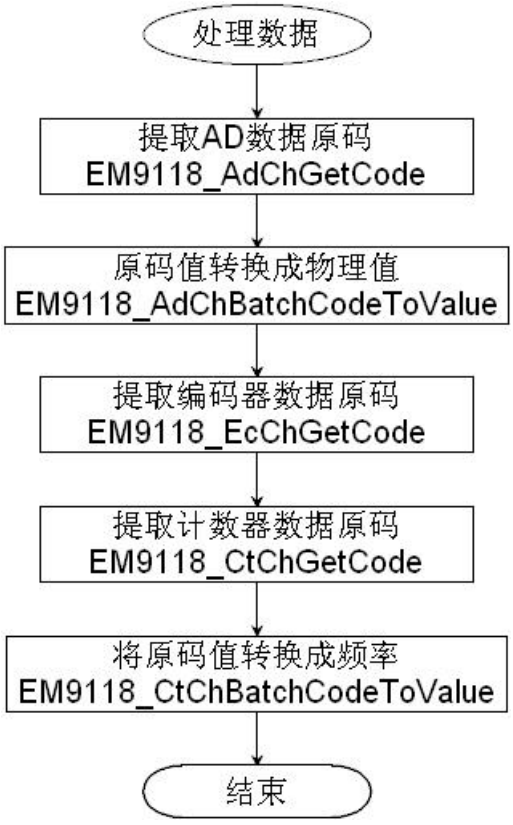
EM9118_AdChIsInFifo
EM9118_CtChIsInFifo
EM9118_EcChIsInFifo
EM9118_HcReadData
EM9118_AdChGetCode
EM9118_CtChGetCode
EM9118_EcChGetCode

1.7.2 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：


开始
创建设备
EM9118_DeviceCreate
建立命令连接
EM9118_CmdConnect
建立数据连接
EM9118_DataConnect
采集参数设置
（详见右侧）
启动采集
EM9118_HcStart
读取组字节数整数倍字节数的数据
EM9118_HcReadData
处理数据
（详见后页）
进一步处理数据
（存盘或者分析）
是否结束
否
是
停止采集
EM9118_HcStop
关闭数据连接
EM9118_DataClose
关闭命令连接
EM9118_CmdClose
释放设备资源
EM9118_DeviceClose
结束
采集参数设置
设定组采集频率
EM9118_HcSetGroupFreq
设定进入缓冲区的AD
EM9118_AdChlsInFifo
设定进入缓冲区的编码器
EM9118_EcChlsInFifo
设定进入缓冲区的计数器
EM9118_CtChlsInFifo
获得每组字节数
EM9118_GetFifoGroupByteCount
设置AD范围
EM9118_AdSetRange
设置计数器工作方式
EM9118_CtSetModeAll
设置测频基准时钟周期
EM9118_CtSetFreqBase
结束


处理数据
提取AD数据原码
EM9118_AdChGetCode
原码值转换成物理值
EM9118_AdChBatchCodeToValue
提取编码器数据原码
EM9118_EcChGetCode
提取计数器数据原码
EM9118_CtChGetCode
将原码值转换成频率
EM9118_CtChBatchCodeToValue
结束

1.8 读取离线文件编程方法

1.8.1 相关函数：

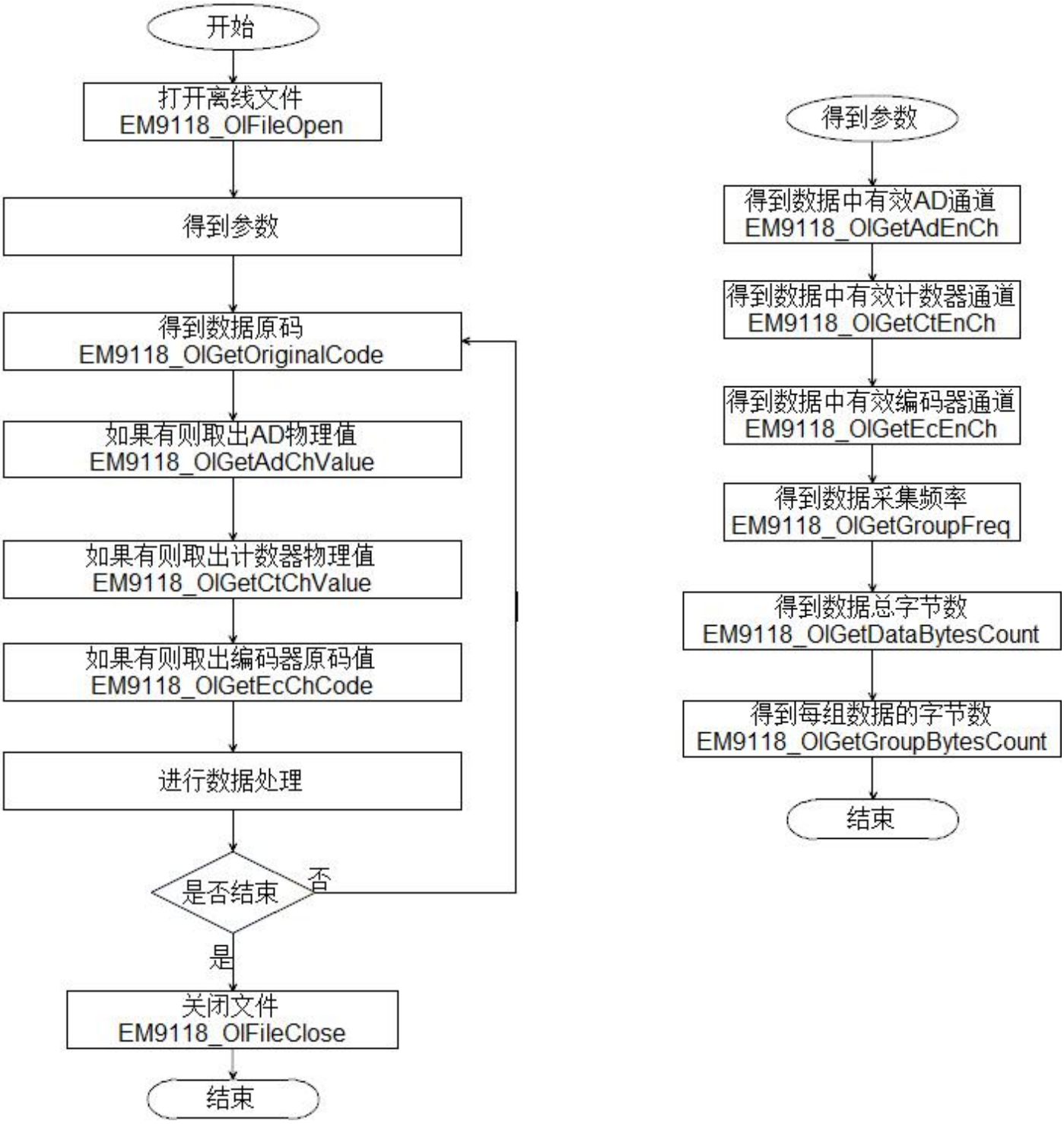
EM9118_OlFileOpen
EM9118_OlGetAdEnCh
EM9118_OlGetCtEnCh
EM9118_OlGetEcEnCh
EM9118_OlGetGroupFreq
EM9118_OlGetDataBytesCount
EM9118_OlGetGroupBytesCount
EM9118_OlGetOriginalCode
EM9118_OlGetAdChValue
EM9118_OlGetCtChCode
EM9118_OlGetEcChCode
EM9118_OlFileClose

1.8.2 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。用户还可以使用 Matlab 等分析软件直接调用本动态库来获得数据后进行分析，也可以调用“ZTHelper.dll”里面提供的函数将各个物理值存成 CSV 格式的表格文件，然后使用 Matlab 读取 CSV 文件进行分析，关于“ZTHelper.dll”里面的函数使用方法，请参考《ZTHelper 编程手册》。CSV 文件是逗号分隔的文本文件，还可以直接使用 Excel 等软件打开转换成带图形的表格文件。

流程图如下：



第二章 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。

2.1 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

2.2 函数说明

请见“EM9118.h”文件

第三章 基本原理

3.1 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。

3.1.1 物理值转换

EM9318B 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。

3.1.1.1 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，每个通道会有一点细微的差别。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：

实际电压值 = （原码值 - 零点值）/（满度值-零点值）* 9(V)

±5V：

实际电压值 = （原码值 - 零点值）/（满度值-零点值）* 4.5(V)

3.1.1.2 计数器输入：32 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”

3.1.1.3 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。

3.1.2 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 60 字节，每个字节含义如下表所示

字节序号	1～2	3～4	5～6	7～8	9～10	11~12	13~14	15~16	17～18
含义	AD1	AD2	AD3	AD4	AD5	AD6	AD7	AD8	AD9
字节序号	19～20	21～22	23～24	25～26	27～28	29～30	31～32	33～34	35～36
含义	AD10	AD11	AD12	AD13	AD14	AD15	AD16	AD17	AD18
字节序号	37～40	41～44	45～48	49～52	53～56	57～60			
含义	CT1	CT2	CT3	CT4	EC1	EC2			

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。

当所有 AD 通道允许进入 FIFO 时，每组字节数为 36，含义如下表：

字节序号	1～2	3～4	5～6	7～8	9～10	11~12	13~14	15~16	17～18
含义	AD1	AD2	AD3	AD4	AD5	AD6	AD7	AD8	AD9
字节序号	19～20	21～22	23～24	25～26	27～28	29～30	31～32	33～34	35～36
含义	AD10	AD11	AD12	AD13	AD14	AD15	AD16	AD17	AD18

当只有 AD1～AD6 允许进入 FIFO 时，每组字节数为 12，含义如下表：

字节序号	1～2	3～4	5～6	7～8	9～10	11~12
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 16，含义如下表：

字节序号	1～4	5～8	9～12	13～16
含义	CT1	CT2	CT3	CT4

3.2 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 36MHz（36,000,000Hz）。因此**测频基准时间**最小值约为 27 纳秒（0.000027 毫秒），由此决定设置的时间只能是 27 纳秒的整数倍。

测频精度的基本公式如下：

测频精度 =1 / 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。

计数器值 = 被测信号频率（赫兹） * 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。

计数器值 = 1 / 低频被测信号频率（赫兹） * 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 豪秒）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。

3.3 PWM 频率输出原理

本设备使用 4MHz 时钟作为 PWM 输出基准，16 位分频，因此最大输出频率：

4000000/65535 ≈ 61.036Hz

频率误差和分频系数有关系，比如在 61Hz 的时候，下一个挡位就是

4000000/65534 ≈ 61.037Hz

此时的精度大概为 1/65535 ≈ 0.0015%，我们使用的晶振是 50ppm 的，因此精度大概是 0.0075%

在较高频率下，主要影响精度的因素就成为了分频系数，比如输出 100KHz 时，需要的分频系数是 40，则分频系数是 39 时候对应的频率就是 102.56KHz。这样当想获得 102.56KHz~100KHz 之间的时候，最大误差就可能是：

1/40/2 * 100% = 12.5%

此时晶振本身的误差就可以忽略不计了。

如果想在更高频率下获得更高的精度，则需要定制更高频率的 PWM 输出基准。

占空比的精度可以仿照上面的计算方法：

占空比精度 =1 / 高电平系数 * 100%

更新记录

时间	更改内容
2015-12-22	初次发布
2016-01-19	修改 EM9118_EcClear 函数说明
2016-03-15	添加了“读取离线文件编程方法”以及相关函数的说明
2017-02-09	修改了一些词语错误
2017-08-29	将“函数说明”小节内容替换成请见头文件
2018-04-26	增加“基本原理”章节
2019-03-08	增加“计数器测频原理”小节
2019-03-21	增加“PWM 频率输出原理”小节
2020-06-22	修正内部采集和高速采集流程图错误