目录

一．舵机 PWM 信号介绍

1．PWM 信号的定义

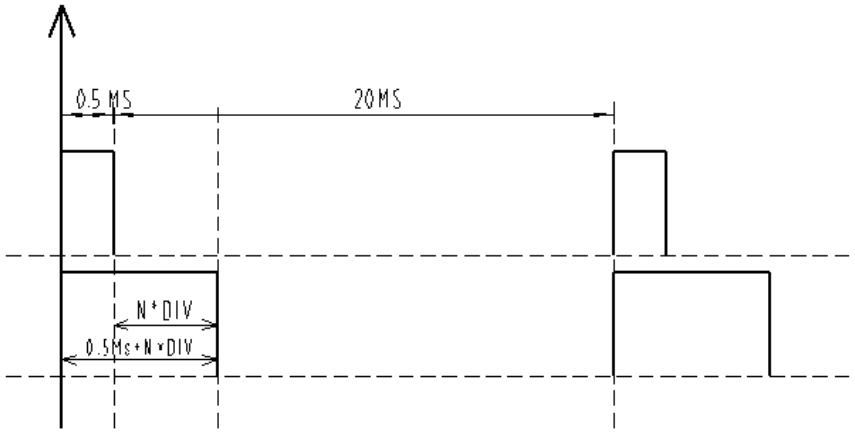
PWM 信号为脉宽调制信号，其特点在于他的上升沿与下降沿之间的时间宽度。具体的时间宽窄协议参考下列讲述。我们目前使用的舵机主要依赖于模型行业的标准协议，随着机器人行业的渐渐独立，有些厂商已经推出全新的舵机协议，这些舵机只能应用于机器人行业，已经不能够应用于传统的模型上面了。

目前，北京汉库的 HG14-M 舵机可能是这个过渡时期的产物，它采用传统的 PWM 协议，优缺点一目了然。优点是已经产业化，成本低，旋转角度大（目前所生产的都可达到 185 度）；缺点是控制比较复杂，毕竟采用 PWM 格式。

但是它是一款数字型的舵机，其对 PWM 信号的要求较低：

（1） 不用随时接收指令，减少 CPU 的疲劳程度；

（2） 可以位置自锁、位置跟踪，这方面超越了普通的步进电机；



N 值为1-250

0.5mS ≤ 0.5Ms+N×DIV ≤ 2.5mS

图 1-1

其 PWM 格式注意的几个要点：

（1） 上升沿最少为 0.5mS，为 0.5mS---2.5mS 之间；

（2） HG14-M 数字舵机下降沿时间没要求，目前采用 0.5Ms 就行；也就是说 PWM 波形可以是一个周期 1mS 的标准方波；

（3） HG0680 为塑料齿轮模拟舵机，其要求连续供给 PWM 信号；它也可以输入一个周期为 1mS 的标准方波，这时表现出来的跟随性能很好、很紧密。

2．PWM 信号控制精度制定

我们采用的是 8 位 AT89C52CPU，其数据分辨率为 256，那么经过舵机极限参数实验，得到应该将其划分为 250 份。

那么 0.5mS---2.5Ms 的宽度为 2mS = 2000uS。

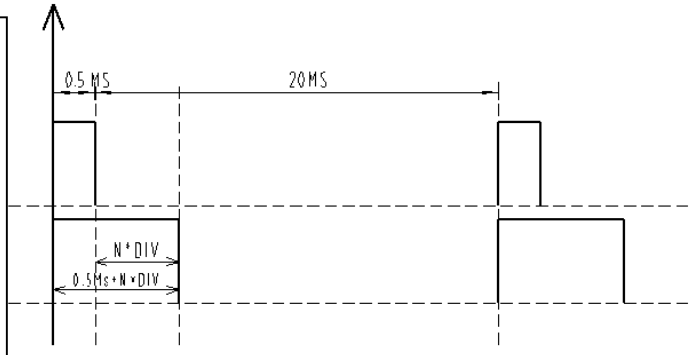
2000uS÷250=8uS

则：**PWM 的控制精度为 8us**

我们可以以 8uS 为单位递增控制舵机转动与定位。

舵机可以转动 185 度，那么 185 度÷250=0.74 度，

则：**舵机的控制精度为 0.74 度**



N值为1-250

0.5mS ≤ 0.5Ms+N×DIV ≤ 2.5mS

图 1-2

1 DIV = 8uS ; 250DIV=2mS

时基寄存器内的数值为：（#01H）01 ----（#0FAH）250。

共 185 度，分为 250 个位置，每个位置叫 1DIV。

则：185÷250 = 0.74 度 / DIV

PWM 上升沿函数： 0.5mS + N×DIV

0uS ≤ N×DIV ≤ 2mS

0.5mS ≤ 0.5Ms+N×DIV ≤ 2.5mS

二．单舵机拖动及调速算法

1．舵机为随动机构

（1）当其未转到目标位置时，将全速向目标位置转动。

（2）当其到达目标位置时，将自动保持该位置。

所以对于数字舵机而言，PWM 信号提供的是目标位置，跟踪运动要靠舵机本身。

（3）像 HG0680 这样的模拟舵机需要时刻供给 PWM 信号，舵机自己不能锁定目标位置。所以我们的控制系统是一个目标规划系统。

（1）HG14-M 舵机的位置控制方法

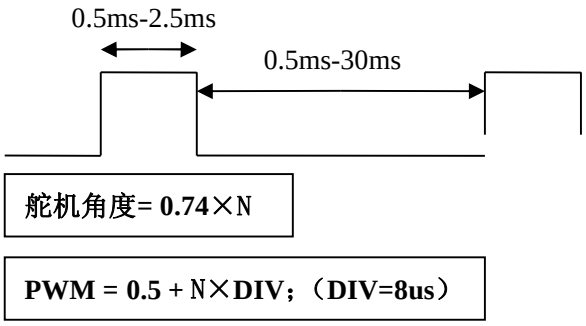
舵机的转角达到 185 度，由于采用 8 为 CPU 控制，所以控制精度最大为 256 份。目前经过实际测试和规划，分了 250 份。具体划分参见《250 份划分原理》。

将 0—185 分为 250 份，每份 0.74 度。

控制所需的 PWM 宽度为 0.5ms—2.5ms，宽度 2ms。

2ms÷250=8us；

所以得出：**PWM 信号 = 1 度/8us**；



舵机角度= 0.74×N

PWM = 0.5 + N×DIV；（DIV=8us）

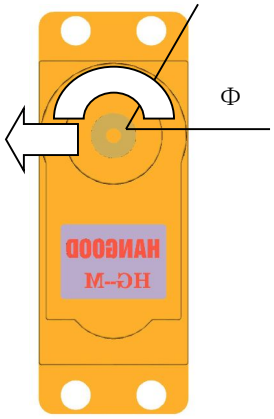
角度	0	45	90	135	180
N	0	3E	7D	BB	FA
PWM	0.5ms	1ms	1.5ms	2ms	2.5ms

（2）HG14-M 舵机的运动协议

舵机的转动方向为：
逆时针为正转

Φ对应 N 值
N=#00H，Φ=0 度
N=#F5H，Φ=180 度

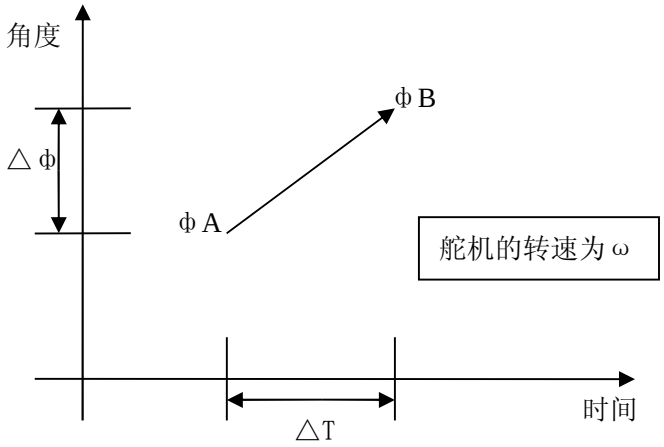
1 ≤ N ≤ 245



运动时可以外接较大的转动负载，舵机输出扭矩较大，而且抗抖动性很好，电位器的线性度较高，达到极限位置时也不会偏离目标。

2. 目标规划系统的特征

（1）舵机的追随特性



① 舵机稳定在 A 点不动；
② CPU 发出 B 点位置坐标的 PWM 信号；
③ 舵机全速由 A 点转向 B 点；

△ф = фB - фA	△T = △ф ÷ ω

④ CPU 发出 B 点 PWM 信号后，应该等待一段时间，利用此时间舵机才能转动至 B 点。

那么，具体的保持（等待）时间如何来计算，如下讲解：

令：保持时间为 Tw

当 Tw ≥ △T 时，舵机能够到达目标，并有剩余时间；

当 Tw ≤ △T 时，舵机不能到达目标；

理论上：当 Tw=△T 时，系统最连贯，而且舵机运动的最快。

实际过程中由于 2 个因素：

① 1 个机器人身上有多个舵机，负载个不相同，所以ω不同；

② 某个舵机在不同时刻的外界环境负载也不同，所以ω不同；

则连贯运动时的极限△T 难以计算出来。

目前采取的方法是经验选取ω值。

（2）舵机ω值测定

舵机的ω值随时变化，所以只能测定一个平均值，或称出现概率最高的点。

依据　① 厂商的经验值；

② 采用 HG14-M 具体进行测试；

测试实验：① 将 CPU 开通，并开始延时 Tw；

② 当延时 Tw 到达后，观察舵机是否到达目标；

测定时采用一段双摆程序，伴随示波器用肉眼观察 Tw 与△T 的关系。

（3）舵机ω值计算

一般舵机定为 0.16--0.22 秒/60 度；

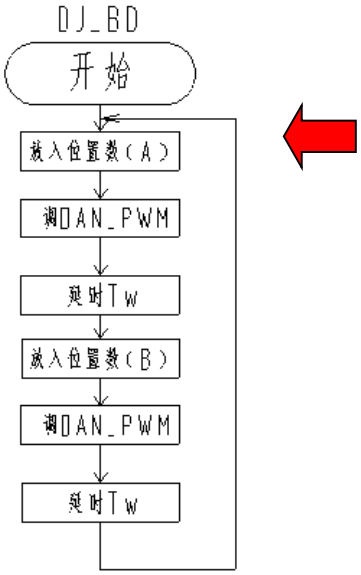
取 0.2 秒/60 度 ＞＞ 1.2 秒/360 度 ＞＞ 0.617 秒/185 度

则ω为 360 度/1.2 秒，2Π/1.2 秒

ω=300 度/秒

那么 185 度转动的时间为 185 度÷360 度/1.2 秒 = 0.6167 秒。

（4）采用双摆试验验证



观察实验过程中的 Tw 与△T 的关系

发现：当 Tw 定在 0.618 秒时，利用示波器观察到舵机能够运动至 2 个目标点。

则：Tw=△T= 0.618 秒

实验过程中，设定舵机运转的目标角度查为 185 度。

最终：ω=0.2 秒/60 度

3．DAV 的定义

将 185 度的转角分为 250 个平均小份。
则：每小份为 0.74 度。

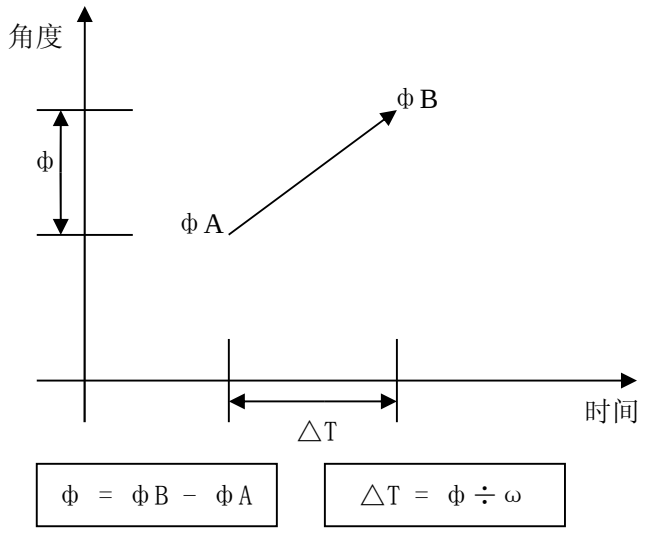
定义如下：DAV = 0.74 度

由于： ω = 0.2 秒/60 度
则：运行 1 DAV 所需时间为：0.72 度÷0.2 秒/60 度 = 2.4 mS；

4．DIV 的定义

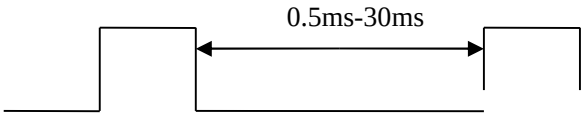
舵机电路支持的 PWM 信号为 0.5mS—2.5mS，总间隔为 2mS。
若分为 250 小份，则 2mS÷250 = 0.008 mS = 8uS

定义如下：DIV = 8uS



那么 1 DAV（0.74 度）对应的△T 为：0.74 度÷60 度/0.2 秒 =2.467mS.。

5．单舵机调速算法

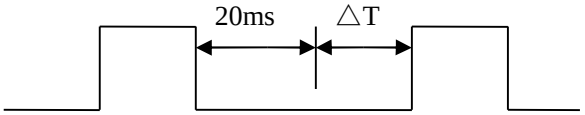


测试内容：将后部下降沿的时间拉至 30ms 没有问题，舵机照样工作。
将后部下降沿的时间拉至 10ms 没有问题，舵机照样工作。
将后部下降沿的时间拉至 2.6ms 没有问题，舵机照样工作。
将后部下降沿的时间拉至 500us 没有问题，舵机照样工作。

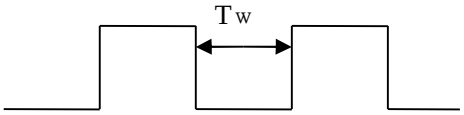
实践检验出：下降沿时间参数可以做的很小。目前实验降至 500uS，依然工作正常。

原因是：（1）舵机电路自动检测上升沿，遇上升沿就触发，以此监测 PWM 脉宽“头”。
（2）舵机电路自动检测下降沿，遇下降沿就触发，以此监测 PWM 脉宽“尾”。

（1）舵机转动时的极限下降沿 PWM 脉宽

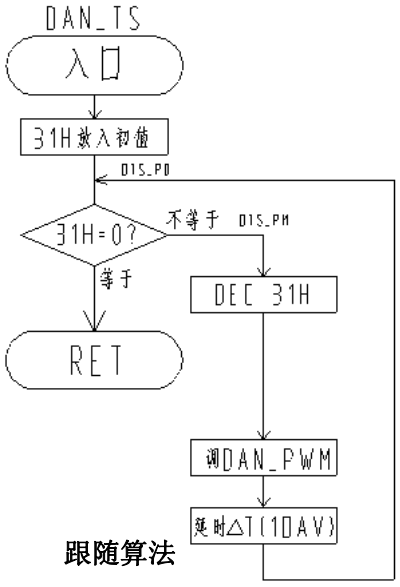


△T：舵机运转 1DAV（7.4 度）所需要的最小时间，目前计算出的数值为 2.467mS；
△T 前面的 20 mS 等待时间可以省略，舵机依然工作；而且得出舵机跟随的最快驱动方式。



极限转动方式

实验得出　1.1ms ≤Tw≤ 50ms； 具体实验数据参照下表

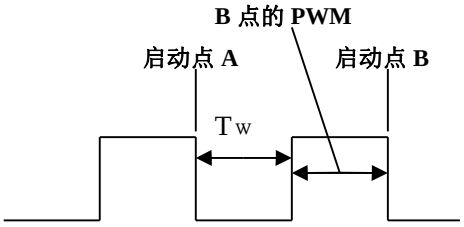


跟随算法

舵机 Tw数据实验表格

Tw值	舵机运转特性	Tw与△T 关系	该程序可行度	备注
500us	不能跟随	Tw< △T	不可行	
800us	不能跟随	Tw< △T	不可行	
1ms	不能跟随	Tw< △T	不可行	
1.1ms	跟随	Tw≈ △T	可行	最快、平滑
1.2ms	跟随	Tw> △T	可行	最快、平滑
1.6ms	跟随	Tw> △T	可行	最快、平滑
2ms	跟随	Tw> △T	可行	最快、平滑
2.6ms	跟随	Tw> △T	可行	最快、平滑
10ms	跟随	Tw>> △T	可行	较慢、平滑
20ms	跟随	Tw>> △T	可行	较慢、平滑
30ms	跟随	Tw>> △T	可行	较慢、平滑
40ms	跟随	Tw>> △T	可以	较慢、微抖
50ms	跟随	Tw>> △T	可以	很慢、微抖
70ms	跟随	Tw>> △T	不可以	很慢、较抖
100ms	跟随	Tw>> △T	不可以	很慢、较抖

令人质疑的地方为 1.1ms 时的表现，得出的 Tw≈ △T；
也就是说 1.1ms = 2.467ms，显然存在问题。
经过考虑重新观察 PWM 波形图发现，电机真正的启动点如下图：



极限转动方式

实际上由 A 到 B 的运动时间为：△T = Tw +（B 点的）PWM

三．8 舵机联动单周期 PWM 指令算法

1．控制要求

要求同时发给 8 个舵机位置目标值，该指令的执行周期尽量短，目的有 2 个：
其一，是为了将来扩充至 24 舵机；其二，目标越快，舵机的转动速度越快；
我们以 8 路为 1 组或称 1 个单位，连续发出目标位置，形成连续的目标规划曲线，电机在跟随过程中自然形成了位置与速度的双指标曲线，实现 8 路舵机联动。

2．注意事项

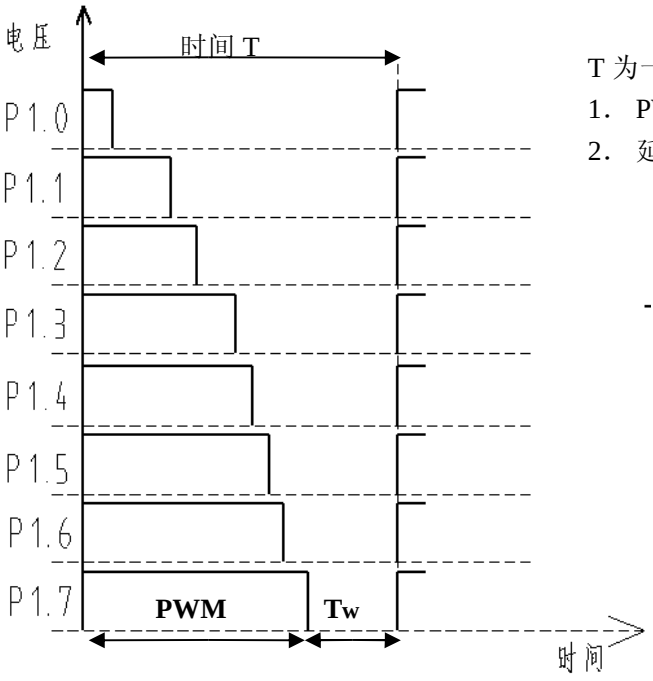
从 24 个端口，P0.0、P1.0 到 P2.0，单 DIV 循环的最小时间只有 8us，所以串行运算是不行的，那么就采用并行运算。

目前采用的并行算法是 P0.0—P0.7 为一个基本单位，8 位一并。

实际案例：P1 口的 8 个位置个不相同；

端口	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
N 寄存器	37H	36H	35H	34H	33H	32H	31H	30H
目标位置（度）	180	135	90	60	50	45	0.74	0
N 数值（整数）	250	187．5	125	81．1	67．6	62．5	1	0
PWM 宽度 ms	2.500	2.000	1.500	1.148	1.041	1.000	0.508	0.500

注意：N 为整数，依照上表看出，由于整数原因，定位不能实现的有 45 度、60 度等。



T 为一个周期，分为 2 部分
1． PWM 宽
2． 延时等待

3．8 路 PWM 信号发生算法解析

我们预计将整个周期控制在 3.5-5ms 内；
由上图得知：P1 口的 8 个端在不同时间产生下降沿。
那么由上例如：我们的 P1.5 口，他的 N 为 125
那么就需要它在 125 个 DIV 后产生下降沿，时间为（125*8us=1000us）。
我们在其中发现 2 个关键参数：①时间参数 N=125
②逻辑参数 P1.5=#0DFH
逻辑参数的定义：如下，采用 ANL 指令，操作 P1 口。

ANL 端口逻辑参数表

	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	备注
P1.0= # FEH	1	1	1	1	1	1	1	0	
P1.1= # FDH	1	1	1	1	1	1	0	1	
P1.2= # FBH	1	1	1	1	1	0	1	1	
P1.3= # F7H	1	1	1	1	0	1	1	1	
P1.4= # EFH	1	1	1	0	1	1	1	1	
P1.5= # DFH	1	1	0	1	1	1	1	1	
P1.6= # BFH	1	0	1	1	1	1	1	1	
P1.7= # 7FH	0	1	1	1	1	1	1	1	

例如：将 P1.5 口产生下降沿，就将# 0DFH 去“ANL” P1 口。
逻辑“ANL”指令，冯“0”得“0”，不影响其他位。

具体的程序操作如下：
① 开 3.5ms 定时中断
② 取出 8 个端（P1.0-P1.7）的位置值，也就是 8 个 N 值；并赋予相应的端逻辑参数；
③ 将这 8 个值由大到小排列，相应端的逻辑参数值也随着 N 的顺序排列，一一对应；
④ 将 N 值做减法，求得： M1=N1 M5=N5-N4
M2=N2-N1 M6=N6-N5
M3=N3-N2 M7=N7-N6
M4=N4-N3 M8=N8-N7

⑤ 取出 M1，延时 M1*DIV，ANL 相应的逻辑参数；
取出 M2，延时 M2*DIV，ANL 相应的逻辑参数；
取出 M3，延时 M3*DIV，ANL 相应的逻辑参数；
取出 M4，延时 M4*DIV，ANL 相应的逻辑参数；
取出 M5，延时 M5*DIV，ANL 相应的逻辑参数；
取出 M6，延时 M6*DIV，ANL 相应的逻辑参数；
取出 M7，延时 M7*DIV，ANL 相应的逻辑参数；
取出 M8，延时 M8*DIV，ANL 相应的逻辑参数；

⑥ 8 个端的下降沿全部产生完毕，等待一定的 **Tw** 值，或等待 3.5ms 中断的到来；
⑦ 中断到来后，清理中断标志，然后结束该程序。RET

注意事项：当进行逐个排序延时的过程中，CPU 要取出 M1、M2、M3….M8，那么会有 1 个取数指令周期，当 CPU 采用 12MHz 时为 1us。最终应该在第 8 个延时，即 M8 时扣除掉，具体指令参见指令集。

4．N 排序子程序 RAM 的制定

入口处

	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
N 值寄存器地址	37H	36H	35H	34H	33H	32H	31H	30H
ANL 逻辑数寄存器地址	3FH	3EH	3DH	3CH	3BH	3AH	39H	38H
ANL 逻辑数值	#7FH	#BFH	#DFH	#EFH	#F7H	#FBH	#FDH	#FEH

备注：37 寄存器内存放的是 P1.7 端口的 N 值；3F 寄存器内存放的是 P1.7 端口的 ANL 逻辑参数值；

出口处

从左到右为 N 值从大到小排列　　（大 ＞ N 值 ＞ 小）

N 值寄存器地址	30H	31H	32H	33H	34H	35H	36H	37H
ANL 逻辑数寄存器地址	38H	39H	3AH	3BH	3CH	3DH	3EH	3FH
ANL 逻辑数值	未知	未知	未知	未知	未知	未知	未知	未知

所谓的“未知”：由于排列按照大到小顺序，“未知”内存放的为端口信息要根据排序做相应的调整。

备注：30H 内存放的是某位的 N 值，其值最大；
37H 内存放的是某位的 N 值，其值最小；
38H—3FH 内存放 ANL 数，可以根据其数值判断出是具体那个端口的下降沿。
例如：其值为“#FBH”那么它就是 P1.2；

5．N 差子程序解析

所谓 N 差子程序，要观察 PWM 口的逻辑时序特性。要求连续将 8 位端口分别产生下降沿。所以有个先后问题，解决的方法有 2 种：

①打开 8 个时间中断；

②按先后顺序排列，先后触发；

由于 CPU 不能开启 8 个中断，所以采用后者方法，那么，就可以得出以下结论：

第 1 个 触发位所用的时间为 N1- 0 =M1；
第 2 个 触发位所用的时间为 N2-N1=M2；
第 3 个 触发位所用的时间为 N3-N2=M3；
第 4 个 触发位所用的时间为 N4-N3=M4；
第 5 个 触发位所用的时间为 N5-N4=M5；
第 6 个 触发位所用的时间为 N6-N5=M6；
第 7 个 触发位所用的时间为 N7-N6=M7；
第 8 个 触发位所用的时间为 N8-N7=M8；

大 小

入口：30H 31H 32H 33H 34H 35H 36H 37H

由于上接排序字程序，所以已经按照从大到小排列，做减法后差所以全为正数。

大 - 小

30H - 31H ⇨ 30H
31H - 32H ⇨ 31H
32H - 33H ⇨ 32H
33H - 34H ⇨ 33H
34H - 35H ⇨ 34H
35H - 36H ⇨ 35H
36H - 37H ⇨ 36H
37H ⇨ 37H

出口	30H	31H	32H	33H	34H	35H	36H	37H
	差	差	差	差	差	差	差	原数

调用延时程序时，37H 最先出，30H 最后出。

6．关于扫尾问题

（1）提出扫尾的概念

我们提出了 1 个扫尾的新概念：当 CPU 执行完 8 个位的下降沿操作后（最多为 2.5ms），会有向下 1 个周期过渡的时间间隔，其主要为 2 个功能：

①保证下降沿的准确性；

②为舵机的跟踪留出足够的时间；

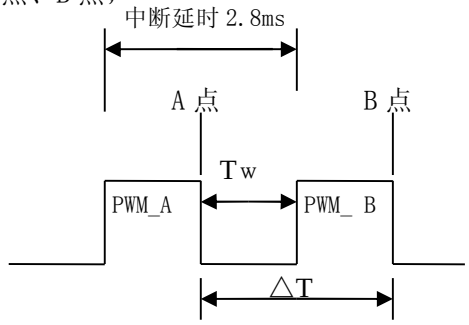
当 PWM 信号以最小变化量即（1DIV=8us）依次变化时，舵机的分辨率最高，但是速度会减慢。例如：先发一个 PWM 信号 N=125，相隔 20ms 后再发 1 个 PWM 信号 N=126。那么舵机在 20ms 内转动了 0.74 度，计算得出：ω=0.74 度/20ms = 37 度/秒；

HG14-M 舵机空载时：ω=300 度/秒

发现与最快速度相差 8 倍之多！

（2）扫尾值的计算

图中有 A 点、B 点，



舵机从 PWM_A 发出后开始转动，经过△T 时间后接收完毕 PWM_B 信号后，又重新开始新的转动。

PWM 处在最小极限长度时：

PWM_A = 0.5ms

PWM_B = 0.5ms

必要条件：△T≥2.467ms

∵Tw=△T - PWM_ B

∴limtTw=2.467ms-0.5ms= 1.967ms

PWM 处在最长极限长度时：

PWM_A = 2.5ms

PWM_B = 2.5ms

必要条件：△T≥2.467ms

∵Tw=△T - PWM_ B

∴limtTw=2.467ms-2.5ms= -0.033ms

为了保证在 2 种极限情况下舵机都能正常工作，我们取个较长的延时，其经验值为 2.8ms；这样舵机都能正常跟随而且速度接近最大值，采用中断法延时 2.8ms。