

对于大多数 MCU 来说，浮点运算、三角函数永远是可望不可及的。说他可望是因为在编译环境函数库中，这些运算都可以通过软件来实现，但他付出的代价是较长代码长度和大量的 CPU 运算时间。代码长度也许是系统可以承受的，但大量的 CPU 运算时间则是实时控制系统无法接受的。

在 Xinnova ( www.xinnovatech.com ) 新推出的一系列低成本、高性能划时代 MCU 产品 XN62L 和 XN12L 中，三角函数得到了彻底解决。它通过集成高达 32 位 CORDIC 矢量计算机，可以轻易并高速实现复杂三角函数在各种实时系统上的矢量控制算法。如直流无刷电机 ( BLDC ) 矢量控制，步进电机的细分控制，逆变控制和坐标系的旋转计算。

Xinnova CORDIC 的软件接口非常简单：一个控制寄存器，三个计算输入寄存器 X,Y,PH 和三个计算结果寄存器 RLT\_X, RLT\_Y, RLT\_PH。

使用步骤如下：

1. 设置 CORDIC 计算精度：CORDIC 支持高达 32 位到计算精度。计算精度与时间成正比。该 CORDIC 精度从 16 到 32 位。一旦控制寄存器的 RES 位被设置的话，所有计算输入寄存器 X, Y 和 PH 寄存器也必须使用相同的位数。

2. 针对不同的计算要求，填入相应的 X, Y 和 PH 值

Phase( 相位 )寄存器输入值范围: CORDIC 只接受从  $-\pi/2$  到  $+\pi/2$  的对应相位值。 $-\pi/2$  到  $+\pi/2$  的相位被映射到  $-0x6487ED51$  到  $+0x6487ED51$  的 32 位精度数或  $-0x6487ED51 >> 16$  到  $+0x6487ED51 >> 16$  的 16 位精度数。

CORDIC x, y 寄存器转换: 只有精度范围内的有效值参与计算(寄存器[RES:0])，其中 RES 位是符号位。在这里，-1 到 +1 同样被映射到有效的寄存器位。例如：如果使用 32 位精度 CORDIC 运算， $-0x7FFF FFFF$  被映射到 -1， $+0x7FFF FFFF$  被映射到 +1。

3. 设置 CORDIC 模式：CORDIC 支持 Sin, Cos 和 Arctan 三种计算。根据要求设置相应的模式。

4. Sin 计算:

- a) 写 PH
- b) 选择 CORDIC 模式 10，Sin/Cos 计算
- c) 读计算结果 RLT\_Y

5. Cos 计算:

- a) 写 PH

b) 选择 CORDIC 模式 10， Sin/Cos 计算

c) 读计算结果 RLT\_X

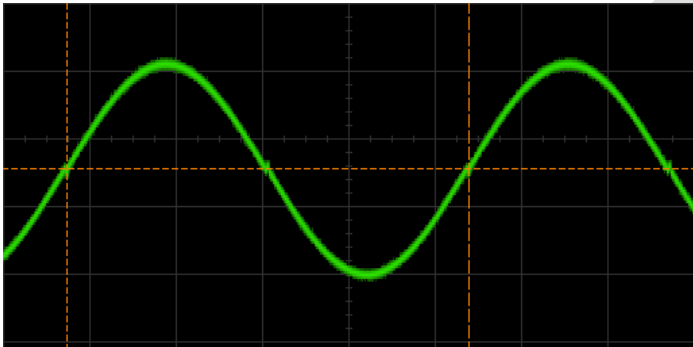
6. Arctan 计算:

a) 写 X 和 Y

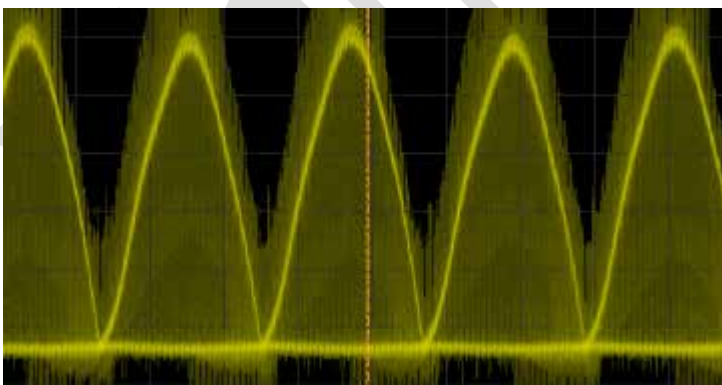
b) 选择 CORDIC 模式 01， Arctan 计算

c) 读计算结果 RLT\_PH

如下是采用 CORDIC 运算后在 DA 端口输出的正弦波。



另一个例子来自使用 CORDIC 做步进电机细分控制的电流波形。



总结：

CORDIC 实现三角函数运算是一种低成本方案。它即可满足嵌入式对矢量运算的要求，又能节约矢量计算成本，是 MCU 实现矢量控制的不二选择。