接触DSP和无刷电机源码时，对于计算的要求比较高。对Q格式做一下了解，当然也解答了我以前的疑问。
1.什么是定点数?
2.印象中的dsp不是应该支持浮点数的运算么?

在看ST的变换源码时，本没了解什么是Q格式，不过当时的理解是将sin(theta)在0-90度的值[0-1]区间做了放大，即为0-32768。下一步计算完后再右移15位缩小。

- 事实也如此**Q格式，就是将一个小数放大若干倍后，用整数来表示小数**

位数系统

做为较为底层的c程序员，或者说计算机的常识，我们对于一个数如何在计算机中存储的。不管是什么类型的数，有无符号，整数小数，到最后都是01010的二进制。整数还好理解，一般的float浮点型数，一位符号位，8位指数位，23位尾数部分，在存取时会多出计算开销。

物理量时间、电压等等，都是类比的、连续的。数位系统，讯号是在不连续的时间点取样，物理量或讯号的大小也不再是连续，而是被量化(Quantized)。在数位系统中，只能用有限字元长度的数字去表示数量的大小，而不能以无限精确的数值(实数)去表示。使用了定点数与浮点数的表示法。

1.定点数(Fixed Point Number)：指一个数字的表示，其小数点是在固定的位置(位元)。
2.浮点数(Floating Point Number)：使用假数以及指数两部分来表示数值。

定点数-Q格式

Q格式是二进制的定点数格式，指定了相应的小数位数和整数位数，Q格式表示为：Qm.n，表示数据用m比特表示整数部分，n比特表示小数部分，共需要m+n+1位来表示这个数据，多余的一位用作符合位。

目的：在没有浮点运算的平台上，更快的处理浮点数据。

Q15

在两个字节下，Q15表示小数位有15位，最高位表示符号位。则表示的范围：[-1<X<0.9999695].

15位即为2的15次方为32768.所以我们将1放大了32768倍。其精度即为1/32768=0.000030517578125。
因为0x7fff为32767，所以正数最大就是0.9999695。

在四个字节下的，第一位表示符号位，16位表示整数为，15位表示小数位。Q15的范围则为[-65536，65535.9999695]

转换

- **整数A->Qx值化**
  即为A*2的x次方
  Q7:A*128

```
1 #define IQ7(A) ((int16_t)((A)*128))
```

- **小数A->Qx值化**
  即为A*2的x次方
  Q7:A*128.0f

```
1 #define FQ7(A) ((int16_t)((A)*128.0f))
```

- **Qx的A int化**
  即为A>>X
  Q7:A>>7

```
1 #define Q7I(A) ((int16_t)((A)>>7))
```

- **Qx的A float化**
  即为A*精度(1/2的x次方)
  Q1: A*5.0000000E-01
  …
  Q7: A*7.8125000E-03

```
1 #define Q7F(A)  ((A)*7.8125000E-03)
```

计算

- **加法-减法**
  加法和减法需要两个Q格式的数据定标相同。直接相加即可。（注意安全的话，需要进行溢出检测）
- **乘法**

```
1 #define Q7_Mul(A,B)   ((A)*(B)>>7)
```

（考虑安全的话，需要考虑溢出的情况）

- **平方**

```
1 #define Q7_Squ(A) ((A)*(A)>>7)
```

（考虑安全的话，需要考虑溢出的情况）

Q格式的运算

1. 定点加减法：须转换成相同的Q格式才能加减
2. 定点乘法：不同Q格式的数据相乘，相当于Q值相加，即Q15数据乘以Q10数据后的结果是Q25格式的数据
3. 定点除法：不同Q格式的数据相除，相当于Q值相减

4. 定点左移：左移相当于Q值增加

5. 定点右移：右移相当于Q减少

Qmath

TI的sdp总提供的库。IQmath一共提供了30种Q格式，具体选择格式要兼顾精度和值.提供了相应的转换和很多数学计算的库

```
long _IQmpyI32int(A, B) //N*long IQ乘long 返回整数部分
long _IQmpyI32frac(A, B)//N*long IQ乘long 返回小数部分
_IQmpy(A, B)            //N*N乘法
_IQrmpy(A, B)           //N*N四舍五入的乘法最后保存结果前(四舍五入)
_IQrsmpy(A, B)          //N*N四舍五入的饱和处理乘法(如果Q26[-32,+32],如果相乘结果超过也会限制到i
_IQmpyI32(A, B)         //N*long IQ乘long
_IQmpyIQX(A, A1, B, B1) //N1*N2两个不同的Q格式乘法,返回全局Q格式
_IQdiv(A, B)            // N/N iq除法

三角函数
_IQsin(A)
_IQsinPU(A)             //正弦函数(标幺值),你占这个圆周的几分之几为单位如果sin((0.25*PI)/(2*PI)
_IQcos(A)
_IQcosPU(A)
_IQatan2(A, B)           //第四象限反正切 tan-1(sin, cos)
_IQatan2PU(A, B)         //第四象限反正切 tan-1(sin, cos)
_IQatan(A, B)           //定点反正切    tan-1(1),,1=sin/cos

_IQNsin(A)
_IQNsinPU(A)            //正弦函数(标幺值),你占这个圆周的几分之几为单位如果sin((0.25*PI)/(2*PI)
_IQNcos(A)
_IQNcosPU(A)
_IQNatan2(_iqA, B)      //第四象限反正切 tan-1(sin, cos)
_IQNatan2PU(_iqA, B)    //第四象限反正切 tan-1(sin, cos)
_IQNatan(A, B)          //定点反正切    tan-1(1),,1=sin/cos

数学函数
_IQNsqrt(A)             //平方根         a^0.5
_IQNisqrt(A)            //平方根倒数     1/a^0.5
_IQNmag(A, B)           //求模运算(sqrt(A^2 + B^2)

_IQsqrt(A)              //平方根         a^0.5
_IQisqrt(A)             //平方根倒数     1/a^0.5
_IQmag(A, B)            //求模运算(sqrt(A^2 + B^2)

其它函数
_IQsat(A, long P, long N)//IQ数值的限幅函数 把A限制到[N P]之间
_IQNabs(A)              //IQ数据的绝对值     |A|
_IQabs(A)               //IQ数据的绝对值     |A|
```