

FSin.bas

Introduction

`fSin` is the basis for the alternatives, since `COS(x)` can be calculated from `SIN(x)` and `TAN(x)` from `COS(x)` and `SIN(x)`.

The functions should be accurate to about 0.25%, and significantly faster. If you need a lot of trig in your code, and it doesn't need to be pinpoint accuracy, these are good alternatives. Note that they more or less acknowledge they are less accurate by returning values of type `Fixed` instead of type `Float`. I did this because it should be fine for the actual accuracy returned, and `Fixed` numbers process faster and smaller than `Float` ones.

- Note that you need only include `fSin` if you only want Sines, but you need `fSin` to use `fCos` or `fTan`.
- Note that these functions use degrees, not radians.

SINE Function

```

FUNCTION fSin(num as FIXED) as FIXED
DIM quad as byte
DIM est1,dif as uByte

num = num MOD 360
'This change made now that MOD works with FIXED types.
'This is much faster than the repeated subtraction method for large angles (much > 360)
'while having some tiny rounding errors that should not significantly affect our results.
'Note that the result may be positive or negative still, and for SIN(360) might come out
'fractionally above 360 (which would cause issues) so the below code still is required.

while num>=360
    num=num-360
end while

while num<0
    num=num+360
end while

IF num>180 then quad=-1
    num=num-180
ELSE quad=1
END IF

IF num>90 then num=180-num

num=num/2
dif=num : rem Cast to byte loses decimal
num=num-dif : rem so this is just the decimal bit

est1=PEEK (@sinetable+dif)
dif=PEEK (@sinetable+dif+1)-est1 : REM this is just the difference to the next up number.

num=est1+(num*dif): REM base +interpolate to the next value.

return (num/255)*quad

sinetable:
asm
DEFB 000,009,018,027,035,044,053,062
DEFB 070,079,087,096,104,112,120,127
DEFB 135,143,150,157,164,171,177,183
DEFB 190,195,201,206,211,216,221,225
DEFB 229,233,236,240,243,245,247,249
DEFB 251,253,254,254,255,255
end asm
END FUNCTION

```

COSINE Function

```

FUNCTION fCos(num as FIXED) as FIXED
    return fSin(90-num)
END FUNCTION

```

TANGENT Function

```

FUNCTION fTan(num as FIXED) as FIXED
    return fSin(num)/fSin(90-num)
END FUNCTION

```