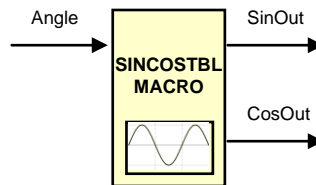


**Description**

This function outputs sin and cos of the input Angle using a discrete look-up table.

**Availability**

This IQ module is available in one interface format:

- 1) The C interface version

**Module Properties**

**Type:** Target Independent, Application Independent

**Target Devices:** 28x Fixed and Floating Point devices

**C Version File Names:** sin\_cos\_table.h

**IQmath library files for C:** IQmathLib.h, IQmath.lib

## C Interface

### C Interface

#### Object Definition

The structure of SINCOSTBL object is defined by following structure definition

```
typedef struct { Uint16 AngleShift;      // Input: scale angle to modulo 512 (length of table)
                Uint16 Angle;          // Input: angle (Q2-Q9)
                _iq SinOut;             // Output: Sin (GLOBAL_Q)
                _iq CosOut;             // Output: Cos (GLOBAL_Q)
            } SINCOSTBL;
```

```
typedef SINCOSTBL *SINCOSTBL_handle;
```

Item	Name	Description	Format	Range(Hex)
Inputs	AngleShift	Used to scale the input angle from modulo 4,8,32,64,128,256 or 512 to modulo 512 (convert Q2-Q9 to Q9)	Q0	0-5
	Angle	Input Angle modulo 4,8,32,64,128,256 or 512 (Q2-Q9)	Q2-Q9	0-3 0-7 0-1F 0-3F 0-7F 0-FF 0-1FF
Outputs	SinOut	Sin of input Angle	GLOBAL_Q	80000000-7FFFFFFF
	CosOut	Cos of input Angle	GLOBAL_Q	80000000-7FFFFFFF

GLOBAL\_Q valued between 1 and 30 is defined in the IQmathLib.h header file.

### Special Constants and Data types

#### SINCOSTBL

The module definition is created as a data type. This makes it convenient to instance an interface to the sin/cos look-up table. To create multiple instances of the module simply declare variables of type SINCOSTBL.

#### SINCOSTBL\_handle

User defined Data type of pointer to SINCOSTBL module

#### SINCOSTBL\_DEFAULTS

Structure symbolic constant to initialize SINCOSTBL module. This provides the initial values to the terminal variables as well as method pointers.

### Methods

#### SINCOSTBL\_MACRO(SINCOSTBL\_handle);

This definition implements one method viz., the sin/cos look-up table macro. The input argument to this macro is the module handle.

## Module Usage

### Instantiation

The following example instances two SINCOSTBL objects  
SINCOSTBL st1, st2;

### Initialization

To Instance pre-initialized objects  
SINCOSTBL st1 = SINCOSTBL\_DEFAULTS;  
SINCOSTBL st2 = SINCOSTBL\_DEFAULTS;

### Invoking the computation macro

SINCOSTBL\_MACRO(st1);  
SINCOSTBL\_MACRO(st2);

## Example

The following pseudo code provides the information about the module usage.

```
main()
{
    st1.AngleShift = 0;           // Input Angle should be modulo 512
                                   // giving 128 microsteps/step
    st2.AngleShift = 2;           // Input Angle should be modulo 128
                                   // giving 32 microsteps/step
}

void interrupt periodic_interrupt_isr()
{
    st1.Angle = ang1;             // Pass input to st1
    st2.Angle = ang2;             // Pass input to st2

    SINCOSTBL_MACRO(st1);         // Call compute macro for st1
    SINCOSTBL_MACRO(st2);         // Call compute macro for st2

    sin1 = st1.SinOut;            // Access the outputs of st1
    cos1 = st1.CosOut;            // Access the outputs of st1

    sin2 = st2.SinOut;            // Access the outputs of st2
    cos2 = st2.CosOut;            // Access the outputs of st2
}
```

## Technical Background

This module implements a discrete look-up table that outputs the sin and cos of an angle. The intended use is as a variable resolution sin/cos generator for microstepping of stepper motors.

The look-up table that is used is part of the IQmathTables contained in the IQmath library. The tables can reside in ROM or RAM depending on the device that is used. The tables are in Q30 format and provide 512 discrete sin and cos points. (See C28x IQmath Library documentation (SPRC990) and the boot ROM reference guide for your device for more information on these tables)

The range of the Angle input varies based on the desired resolution. A 4, 8, 16, 32, 64, 128, 256 or 512 point sin/cos waveform can be generated. The AngleShift setting is used to scale from the desired resolution to the full 512 point resolution of the table. AngleShift should be set as;

$$AngleShift = \log_2(512) - \log_2(desired\_resolution)$$

AngleShift will then be used within the macro to left shift the Angle input up to the full table resolution.

The table is in Q30 format so the macro will convert each output from Q30 to GLOBAL\_Q.