
Chapter 5. Fixed Point Math Functions

5.1 INTRODUCTION

Fixed point library math functions are contained in the files, `libq-omf.a` (standard), and `libq-dsp-omf.a` (DSP), where *omf* will be *coff* or *elf* depending upon the selected object module format. The header file is named `libq.h` and is the same for standard or DSP versions of the library. Linker options, `-lq` (standard and DSP) and `-lq-dsp` (DSP only), must be used when linking the respective libraries.

5.1.1 Assembly Code Applications

A free version of the math functions library and header file is available from the Microchip web site.

5.1.2 C Code Applications

The MPLAB C Compiler for PIC24 MCUs and dsPIC DSCs (formerly MPLAB C30) install directory (`c:\Program Files\Microchip\MPLAB C30`) contains the following subdirectories with library-related files:

- `lib` – standard C library files
- `support\h` – header files for libraries

In addition, there is a file, `ResourceGraphs.pdf`, which contains diagrams of resources used by each function, located in `lib`.

5.1.3 Chapter Organization

This chapter is organized as follows:

- Overview of Fixed Point Data Formats
- Using the Fixed Point Libraries
- `<libq.h>` mathematical functions

16-Bit Language Tools Libraries

5.2 OVERVIEW OF FIXED POINT DATA FORMATS

The integer data is encoded as its two's complement to accommodate both positive and negative numbers in binary format. The two's complement can be represented using integer format or the fractional format.

5.2.1 Integer Format

The integer format data is represented as a signed two's complement value, where the Most Significant bit is defined as a sign bit. The range of an N-bit two's complement integer is -2^{N-1} to $2^{N-1}-1$ with a resolution of 1. For a 16-bit integer, the data range is -32768 (0x8000) to +32767 (0x7FFF) with a resolution of 1. For a 32-bit integer, the data range is -2,147,483,648 (0x8000 0000) to +2,147,483,647 (0x7FFF FFFF) with a resolution of 1.

5.2.2 Fractional Format

The fractional data format (Qn.m) has integral part (n) and fractional part (m) and the Most Significant bit represents the sign, thus consisting of (m+n+1) bits. It represents a signed two's complement value. Qn.m format data has a range of $[-2^n, (2^n-2^{-m})]$ with 2^{-m} resolution.

For the binary representation of an N-bit (m+n+1 bits) number in Qn.m:

$$\underbrace{b_{m+n} b_{m+n-1} \dots b_m}_{N-1} . b_{m-1} \dots b_1 b_0$$

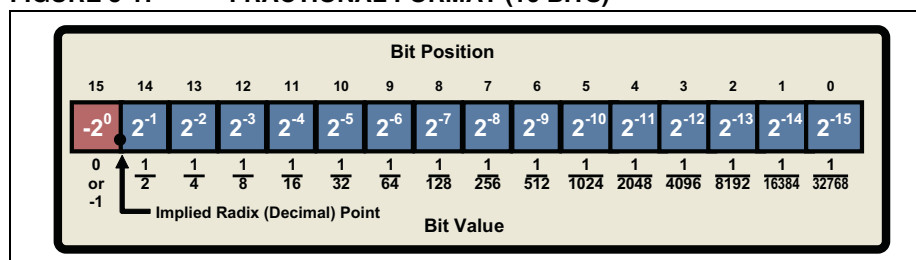
the value is give by the equation:

$$Value = -b_{N-1}2^n + \sum_{l=0}^{N-2} b_l 2^{l-m}$$

5.2.2.1 Q15 (1.15) FORMAT

In Q15 format, the Most Significant bit is defined as a sign bit and the radix point is implied to lie just after the sign bit followed by the fractional value. This format is commonly referred to as 1.15 or Q15 format, where 1 is the number of bits used to represent the integer portion of the number, and 15 is the number of bits used to represent the fractional portion. The range of an N-bit two's complement fraction with this implied radix point is -1.0 to $(1 - 2^{1-N})$. For a 16-bit fraction, the 1.15 data range is -1.0 (0x8000) to +0.999969482 (0x7FFF) with a precision of 3.05176×10^{-5} .

FIGURE 5-1: FRACTIONAL FORMAT (16 BITS)



Fixed Point Math Functions

The following example shows the conversion of a two's complement 16-bit integer +24576 to Q15 value +0.75.

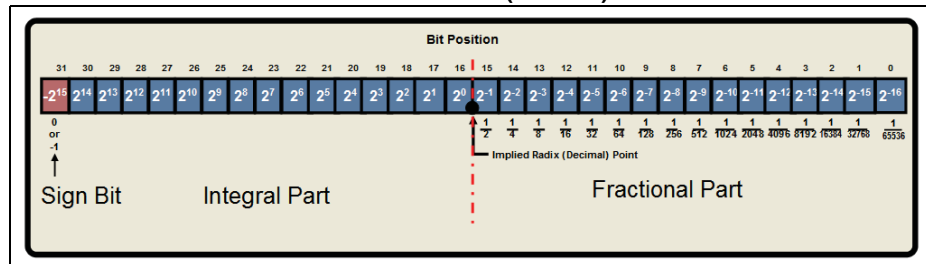
EXAMPLE 5-1: CONVERSION OF A TWO'S COMPLEMENT 16-BIT INTEGER TO Q15

Binary		Dec		Q15	
0	$0 \times (-2^{15})$	0	$0 \times (-2^0)$	0	
			●	●	Radix point
1	1×2^{14}	16384	1×2^{-1}	0.5	
1	1×2^{13}	8192	1×2^{-2}	0.25	
0	0×2^{12}	0	0×2^{-3}	0	
0	0×2^{11}	0	0×2^{-4}	0	
0	0×2^{10}	0	0×2^{-5}	0	
0	0×2^9	0	0×2^{-6}	0	
0	0×2^8	0	0×2^{-7}	0	
0	0×2^7	0	0×2^{-8}	0	
0	0×2^6	0	0×2^{-9}	0	
0	0×2^5	0	0×2^{-10}	0	
0	0×2^4	0	0×2^{-11}	0	
0	0×2^3	0	0×2^{-12}	0	
0	0×2^2	0	0×2^{-13}	0	
0	0×2^1	0	0×2^{-14}	0	
0	0×2^0	0	0×2^{-15}	0	
	Sum	+24576	Sum	+0.75	

5.2.2.2 Q15.16 FORMAT

In the Q15.16 format, the Most Significant bit is defined as a sign bit, followed by 16 bits of the integral part. The radix point is implied to lie just after the integral part, followed by 16 bits of the fractional value. This format is commonly referred to as Q15.16 format. The range of Q15.16 numbers is from -32768.0 (0x8000 0000) to +32767.9999847412109375 (0x7FFF FFFF) and has a precision of 2^{-16} .

FIGURE 5-2: FRACTIONAL FORMAT (32 BITS)



16-Bit Language Tools Libraries

The following example shows the conversion of a two's complement 32-bit integer, -715827882 to Q15.16 value -10922.6666564941.

EXAMPLE 5-2: CONVERSION OF A TWO'S COMPLEMENT 32-BIT INTEGER TO Q15.16

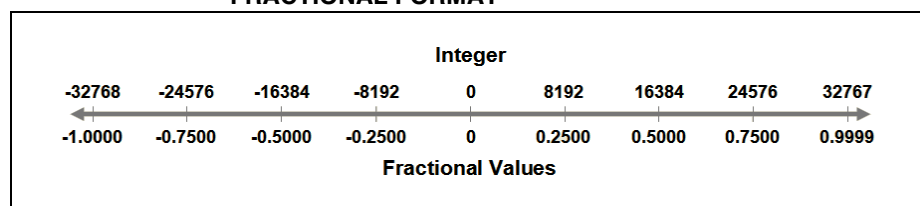
Binary		Dec		Q15.16
1	$1 \times (-2^{31})$	-2147483648	$1 \times (-2^{15})$	-32768
1	1×2^{30}	1073741824	1×2^{14}	16384
0	0×2^{29}	0	0×2^{13}	0
1	1×2^{28}	268435456	1×2^{12}	4096
0	0×2^{27}	0	0×2^{11}	0
1	1×2^{26}	67108864	1×2^{10}	1024
0	0×2^{25}	0	0×2^9	0
1	1×2^{24}	16777216	1×2^8	256
0	0×2^{23}	0	0×2^7	0
1	1×2^{22}	4194304	1×2^6	64
0	0×2^{21}	0	0×2^5	0
1	1×2^{20}	1048576	1×2^4	16
0	0×2^{19}	0	0×2^3	0
1	1×2^{18}	262144	1×2^2	4
0	0×2^{17}	0	0×2^1	0
1	1×2^{16}	65536	1×2^0	1
			●	●
0	0×2^{15}	0	0×2^{-1}	0
1	1×2^{14}	16384	1×2^{-2}	0.25
0	0×2^{13}	0	0×2^{-3}	0
1	1×2^{12}	4096	1×2^{-4}	0.0625
0	0×2^{11}	0	0×2^{-5}	0
1	1×2^{10}	1024	1×2^{-6}	0.015625
0	0×2^9	0	0×2^{-7}	0
1	1×2^8	256	1×2^{-8}	0.00390625
0	0×2^7	0	0×2^{-9}	0
1	1×2^6	64	1×2^{-10}	0.000976563
0	0×2^5	0	0×2^{-11}	0
1	1×2^4	16	1×2^{-12}	0.000244141
0	0×2^3	0	0×2^{-13}	0
1	1×2^2	4	1×2^{-14}	6.10352E-05
1	1×2^1	2	1×2^{-15}	3.05176E-05
0	0×2^0	0	0×2^{-16}	0
	Sum	-715827882	Sum	-10922.6666564941

Radix point

5.2.3 Integer - Fractional Format Mapping

The same two's complement N-bit word may represent an integer format value or a fractional format value. For eg., the 16-bit integer range [-32768, +32767] format maps to Q15 range of [-1.0, +0.999969482]. Figure 5-3 shows the mapping between these formats.

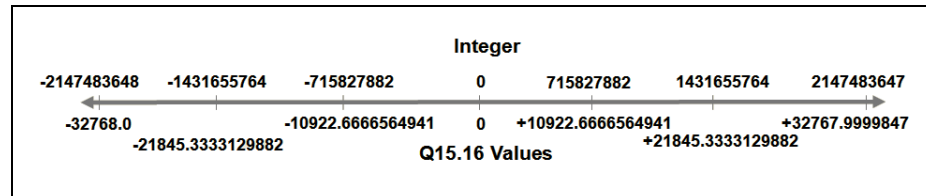
FIGURE 5-3: MAPPING BETWEEN 16-BIT INTEGER FORMAT AND Q15 FRACTIONAL FORMAT



Fixed Point Math Functions

A similar relationship exists between the 32-bit integer format and the Q15.16 format, where the integer range [-2147483648, +2147483647] is mapped to the Q15.16 range [-32768.0, +32767.9999847412109375].

FIGURE 5-4: MAPPING BETWEEN 32-BIT INTEGER FORMAT AND Q15.16 FORMAT



5.2.4 libq Library and Fixed Point Data Format

The functions in the `libq` library use the fixed point data format. The parameters passed and the results generated by the functions are fractional in nature. There are two similar sets of math functions which perform the same math operations. One set supports Q15 operands and the other supports Q15.16 operands. Q15.16 operand functions, naturally, have better precision and range compared to Q15 operand functions.

5.3 USING THE FIXED POINT LIBRARIES

Building an application which utilizes the fixed point libraries requires two types of files: header files and library files.

5.3.1 Header Files

All standard C library entities are declared or defined in one or more standard headers. (See list in **Section 5.1.3 “Chapter Organization”**.) To make use of a library entity in a program, write an include directive that names the relevant standard header.

The contents of a standard header are included by naming them in an include directive, as in:

```
#include <libq.h> /* include fixed point library */
```

The standard headers can be included in any order. Do not include a standard header within a declaration. Do not define macros that have the same names as keywords before including a standard header.

A standard header never includes another standard header.

5.3.2 Library Files

The archived library files contain all the individual object files for each library function. When linking an application, the library file (`libq-omf.a` or `libq-dsp-omf.a`) must be provided as an input to the linker (using the `--library` or `-l` linker option), such that the functions used by the application may be linked into the application. Also, linker options, `-lq` and `-lq-dsp`, must be used when linking the respective libraries.

A typical C application will require three library files: `libc-omf.a`, `libm-omf.a` and `libpic30-omf.a`. (See **Section 1.2 “OMF-Specific Libraries/Start-up Modules”** for more on OMF-specific libraries.) These libraries will be included automatically if linking is performed using the compiler.

16-Bit Language Tools Libraries

5.3.3 Function Naming Conventions

Signed fixed point types are defined as follows:

Qn_m

where:

- n is the number of data bits to the left of the radix point
- m is the number of data bits to the right of the radix point

Note: A sign bit is implied.

For convenience, short names are also defined:

Exact Name	# Bits Required	Short Name
_Q0_15	16	_Q15
_Q15_16	32	_Q16

In this document the terms Q15.16 and Q16 are used interchangeably; however, both imply Q15.16 format. Functions in the library are prefixed with the type of the return value. For example, _Q15acos returns a Q15 value equal to the arc cosine of its argument.

Argument types do not always match the return type. Refer to the function prototype for a specification of its arguments.

In cases where the return value is not a fixed point type, the argument type is appended to the function name. For example, function _itoaQ15 accepts a type Q15 argument.

In cases where two versions of a function are provided, with the same return type but different argument types, the argument type is appended to the function name. For example:

Function Name	Return Type	Argument Type
_Q16reciprocalQ15	_Q16	_Q15
_Q16reciprocalQ16	_Q16	_Q16

Fixed Point Math Functions

5.4 <LIBQ.H> MATHEMATICAL FUNCTIONS

The header file, `libq.h`, consists of macro definitions and various functions that calculate fixed point mathematical operations.

5.4.1 Q15 Functions

Many functions in this section use fixed point Q15 (1.15) format. For each function, the entire range may not be used.

Q15abs

Description:	The function finds the absolute value of a Q15 value.
Include:	<code><libq.h></code>
Prototype:	<code>_Q15 _Q15abs(_Q15 x);</code>
Argument:	<code>x</code> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns the absolute value of <code>x</code> in Q15 format. The value ranges from 0 to 32767.

Q15acos

Description:	This function finds the arc cosine of a Q15 value.
Include:	<code><libq.h></code>
Prototype:	<code>_Q15 _Q15acos(_Q15 x);</code>
Argument:	<code>x</code> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from 17705 to 32767.
Return Value:	This function returns the arc cosine of <code>x</code> in Q15 format. The value ranges from 256 to 32767.

Q15acosByPI

Description:	This function finds the arc cosine of a Q15 value and then divides by PI (π).
Include:	<code><libq.h></code>
Prototype:	<code>_Q15 _Q15acosByPI(_Q15 x);</code>
Argument:	<code>x</code> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns the arc cosine of <code>x</code> , divided by PI, in Q15 format. The value ranges from 82 to 32767.

16-Bit Language Tools Libraries

_Q15add

Description:	This function finds the sum value of two Q15 values. The function takes care of saturation during overflow and underflow occurrences.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15add(_Q15 x, _Q15 y);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767. <i>y</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns the sum of <i>x</i> and <i>y</i> in Q15 format. The value ranges from -32768 to 32767.

_Q15asin

Description:	This function finds the arc sine of a Q15 value.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15asin(_Q15 x);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns the arc sine of <i>x</i> in Q15 format. The value ranges from -32768 to 32767.

_Q15asinByPI

Description:	This function finds the arc sine of a Q15 value and then divides by PI (π).
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15asinByPI(_Q15 x);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns the arc sine of <i>x</i> , divided by PI, in Q15 format. The value ranges from -16384 to 16303.

_Q15atan

Description:	This function finds the arc tangent of a Q15 value.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15atan(_Q15 x);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns the arc tangent of <i>x</i> in Q15 format. The value ranges from -25736 to 25735.

Fixed Point Math Functions

_Q15atanByPI

Description: This function finds the arc tangent of a Q15 value and then divides by PI (π).

Include: `<libq.h>`

Prototype: `_Q15 _Q15atanByPI(_Q15 x);`

Argument: *x* a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.

Return Value: This function returns the arc tangent of *x*, divided by PI, in Q15 format. The value ranges from -8192 to 8192.

_Q15atanYByX

Description: This function finds the arc tangent of a Q15 value divided by a second Q15 value.

Include: `<libq.h>`

Prototype: `_Q15 _Q15atanYByX(_Q15 x, _Q15 y);`

Argument: *x* a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
y a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.

Return Value: This function returns the arc tangent of *y* divided by *x* in Q15 format. The value ranges from -25736 to 25735.

_Q15atanYByXByPI

Description: This function finds the arc tangent of a Q15 value divided by a second Q15 value and then divides the result by PI (π).

Include: `<libq.h>`

Prototype: `_Q15 _Q15atanYByXByPI(_Q15 x, _Q15 y);`

Argument: *x* a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
y a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.

Return Value: This function returns the arc tangent of *y* divided by *x*, divided by PI, in Q15 format. The value ranges from -8192 to 8192.

_Q15atoi

Description: This function takes a string which holds the ASCII representation of decimal digits and converts it into a single Q15 number.
Note: The decimal digit should not be beyond the range: -32768 to 32767.

Include: `<libq.h>`

Prototype: `_Q15 _Q15atoi(const char *s);`

Argument: *s* a buffer holding the ASCII values of each decimal digit.

Return Value: This function returns the integer equivalent of *s* in Q15 format, which range is from -32768 to 32767.

16-Bit Language Tools Libraries

_Q15cos

Description:	This function finds the cosine of a Q15 value.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15cos(_Q15 x);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns the cosine of <i>x</i> in Q15 format. The value ranges from 17705 to 32767.

_Q15cosPI

Description:	This function finds the cosine of PI (π) times a Q15 value.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15cosPI(_Q15 x);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns the cosine of PI times <i>x</i> in Q15 format. The value ranges from -32768 to 32767.

_Q15exp

Description:	This function finds the exponential value of a Q15 value.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15exp(_Q15 x);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 0.
Return Value:	This function returns the exponent value of <i>x</i> in Q15 format. The value ranges from 12055 to 32767.

_Q15ftoi

Description:	This function converts a single precision floating-point value into its corresponding Q15 value.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15ftoi(float x);</code>
Argument:	<i>x</i> a floating-point equivalent number. The corresponding floating-point range is -1 to 0.99996.
Return Value:	This function returns a fixed point number in Q15 format. The value ranges from -32768 to 32767.

Fixed Point Math Functions

_itoaQ15

Description:	This function converts each decimal digit of a Q15 value to its representation in ASCII. For example, 1 is converted to 0x31, which is the ASCII representation of 1.
Include:	<libq.h>
Prototype:	void _itoaQ15(_Q15 x, char *s);
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767. <i>s</i> a buffer holding values in ASCII, at least 8 characters long.
Return Value:	None.

_itofQ15

Description:	This function converts a Q15 value into its corresponding floating-point value.
Include:	<libq.h>
Prototype:	float _itofQ15(_Q15 x);
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns a floating-point equivalent number. The corresponding floating-point range is -1 to 0.99996.

_Q15log

Description:	This function finds the natural log of a Q15 value.
Include:	<libq.h>
Prototype:	_Q15 _Q15log(_Q15 x);
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from 12055 to 32767.
Return Value:	This function returns the natural log of <i>x</i> in Q15 format. The value ranges from -32768 to -1.

_Q15log10

Description:	This function finds the log (base 10) of a Q15 value.
Include:	<libq.h>
Prototype:	_Q15 _Q15log10(_Q15 x);
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from 3277 to 32767.
Return Value:	This function returns the log of <i>x</i> in Q15 format. The value ranges from -32768 to 0.

16-Bit Language Tools Libraries

_Q15neg

Description: This function negates a Q15 value with saturation. The value is saturated in the case where the input is -32768.

Include: `<libq.h>`

Prototype: `_Q15 _Q15neg(_Q15 x);`

Argument: *x* a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.

Return Value: This function returns $-x$ in Q15 format. The value ranges from -32768 to 32767.

_Q15norm

Description: This function finds the normalized value of a Q15 value.

Include: `<libq.h>`

Prototype: `_Q15 _Q15norm(_Q15 x);`

Argument: *x* a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.

Return Value: This function returns the square root of *x* in Q15 format. The value ranges from 16384 to 32767 for a positive number and -32768 to -16384 for a negative number.

_Q15power

Description: This function finds the power result given the base value and the power value in Q15 format.

Include: `<libq.h>`

Prototype: `_Q15 _Q15power(_Q15 x, _Q15 y);`

Argument: *x* a fixed point number in Q15 format, which ranges from 1 to $2^{15}-1$. The value of this argument ranges from 1 to 32767.
y a fixed point number in Q15 format, which ranges from 1 to $2^{15}-1$. The value of this argument ranges from 1 to 32767.

Return Value: This function returns *x* to the power of *y* in Q15 format. The value ranges from 1 to 32767.

_Q15random

Description: This function generates a random number in the range from -32768 to 32767. The random number generation is periodic with period 65536. The function uses the `_Q15randomSeed` variable as a random seed value.

Include: `<libq.h>`

Prototype: `_Q15 _Q15random(void);`

Argument: None.

Return Value: This function returns a random number in Q15 format. The value ranges from -32768 to 32767.

Fixed Point Math Functions

_Q15shl

Description:	This function shifts a Q15 value by <i>num</i> bits, to the left if <i>num</i> is positive or to the right if <i>num</i> is negative. The function takes care of saturating the result, in case of underflow or overflow.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15shl(_Q15 x, short num);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767. <i>num</i> an integer number, which ranges from -15 to 15.
Return Value:	This function returns the shifted value of <i>x</i> in Q15 format. The value ranges from -32768 to 32767.

_Q15shlNoSat

Description:	This function shifts a Q15 value by <i>num</i> bits, to the left if <i>num</i> is positive or to the right if <i>num</i> is negative. The function sets the <code>_Q15shlSatFlag</code> variable in case of underflow or overflow but does not take care of saturation.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15shlNoSat(_Q15 x, short num);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767. <i>num</i> an integer number, which ranges from -15 to 15.
Return Value:	This function returns the shifted value of <i>x</i> in Q15 format. The value ranges from -32768 to 32767.

_Q15shr

Description:	This function shifts a Q15 value by <i>num</i> bits, to the right if <i>num</i> is positive or to the left if <i>num</i> is negative. The function takes care of saturating the result, in case of underflow or overflow.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15shr(_Q15 x, short num);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767. <i>num</i> an integer number, which ranges from -15 to 15.
Return Value:	This function returns the shifted value of <i>x</i> in Q15 format. The value ranges from -32768 to 32767.

_Q15shrNoSat

Description:	This function shifts a Q15 value by <i>num</i> bits, to the right if <i>num</i> is positive or to the left if <i>num</i> is negative. The function sets the <code>_Q15shrSatFlag</code> variable in case of underflow or overflow but does not take care of saturation.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15shrNoSat(_Q15 x, short num);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767. <i>num</i> an integer number, which ranges from -15 to 15.
Return Value:	This function returns the shifted value of <i>x</i> in Q15 format. The value ranges from -32768 to 32767.

16-Bit Language Tools Libraries

_Q15sin

Description:	This function finds the sine of a Q15 value.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15sin(_Q15 x);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns the sine of <i>x</i> in Q15 format. The value ranges from -27573 to 27573.

_Q15sinPI

Description:	This function finds the sine of PI (π) times a Q15 value.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15sinPI(_Q15 x);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns the sine of PI times <i>x</i> in Q15 format. The value ranges from -32768 to 32767.

_Q15sinSeries

Description:	Generates the sine series with the given normalizing frequency, <i>f</i> , and the given number of samples, <i>num</i> , starting from <i>start</i> . Stores the result in buffer, <i>buf</i> .
Include:	<libq.h>
Prototype:	<code>short _Q15sinSeries(_Q15 f, short start, short num, _Q15 *buf);</code>
Argument:	<i>f</i> a fixed point number in Q15 format, which ranges from 0 to $(2^{31}-1)$. The valid range of values for this argument is from -16384 to 16384. The argument represents the Normalizing frequency. <i>start</i> a fixed point number in Q16 format, which ranges from 0 to $(2^{31}-1)$. The valid range of values for this argument is from 1 to 32767. This argument represents the Starting Sample number in the Sine Series. <i>num</i> a fixed point number in Q16 format, which ranges from 0 to $(2^{31}-1)$. The valid range of values for this argument is from 1 to 32767. This argument represents the Number of Sine Samples the function is called to generate. Note: <i>num</i> should not be more than 16383 for dsPIC and 32767 for PIC devices. <i>buf</i> a pointer to the buffer where the generated sine samples would get copied into.
Return Value:	This function returns <i>num</i> , the number of generated sine samples.

Fixed Point Math Functions

_Q15sqrt

Description:	This function finds the square root of a Q15 value.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15sqrt(_Q15 x);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from 1 to 32767.
Return Value:	This function returns the square root of <i>x</i> in Q15 format. The value ranges from 1 to 32767.

_Q15sub

Description:	This function finds the difference of two Q15 values. The function takes care of saturation during overflow and underflow occurrences.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15sub(_Q15 x, _Q15 y);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767. <i>y</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns <i>x</i> minus <i>y</i> in Q15 format. The value ranges from -32768 to 32767.

_Q15tan

Description:	This function finds the tangent of a Q15 value.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15tan(_Q15 x);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -25736 to 25735.
Return Value:	This function returns the tangent of <i>x</i> in Q15 format. The value ranges from -32768 to 32767.

_Q15tanPI

Description:	This function finds the tangent of PI (π) times a Q15 value.
Include:	<libq.h>
Prototype:	<code>_Q15 _Q15tanPI(_Q15 x);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns the tangent of PI times <i>x</i> in Q15 format. The value ranges from -32768 to 32767.

16-Bit Language Tools Libraries

5.4.2 Q16 Functions

Many functions in this section use fixed point Q16 (15.16) format. For each function, the entire range may not be used.

_Q16acos

Description: This function finds the arc cosine of a Q16 value.

Include: `<libq.h>`

Prototype: `_Q16 _Q16acos(_Q16 x);`

Argument: `x` a fixed point number in Q16 format. The value of this argument ranges from -65566 to 65536.

Return Value: This function returns the arc cosine of `x` in Q16 format. The value ranges from -205887 to 205887.

_Q16acosByPI

Description: This function finds the arc cosine of a Q16 value and then divides by PI (π).

Include: `<libq.h>`

Prototype: `_Q16 _Q16acosByPI(_Q16 x);`

Argument: `x` a fixed point number in Q16 format. The value of this argument ranges from -65536 to 65536.

Return Value: This function returns the arc cosine of `x`, divided by PI, in Q16 format. The value ranges from -65536 to 65536.

_Q16asin

Description: This function finds the arc sine of a Q16 value.

Include: `<libq.h>`

Prototype: `_Q16 _Q16asin(_Q16 x);`

Argument: `x` a fixed point number in Q16 format. The value of this argument ranges from -65566 to 65536.

Return Value: This function returns the arc sine of `x` in Q16 format. The value ranges from -102944 to 102944.

_Q16asinByPI

Description: This function finds the arc sine of a Q16 value and then divides by PI (π).

Include: `<libq.h>`

Prototype: `_Q16 _Q16asinByPI(_Q16 x);`

Argument: `x` a fixed point number in Q16 format. The value of this argument ranges from -65536 to 65536.

Return Value: This function returns the arc sine of `x`, divided by PI, in Q16 format. The value ranges from -65536 to 65536.

Fixed Point Math Functions

_Q16atan

Description: This function finds the arc tangent of a Q16 value.

Include: <libq.h>

Prototype: `_Q16 _Q16atan(_Q16 x);`

Argument: *x* a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647.

Return Value: This function returns the arc tangent of *x* in Q16 format. The value ranges from -2147483648 to 2147483647.

_Q16atanByPI

Description: This function finds the arc tangent of a Q16 value and then divides by PI (π).

Include: <libq.h>

Prototype: `_Q16 _Q16atanByPI(_Q16 x);`

Argument: *x* a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647.

Return Value: This function returns the arc tangent of *x*, divided by PI, in Q16 format. The value ranges from -2147483648 to 2147483647.

_Q16atanYByX

Description: This function finds the arc tangent of *y* divided by *x*.

Include: <libq.h>

Prototype: `_Q16 _Q16atanYByX(_Q16 x, _Q16 y);`

Argument: *x* a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647. This forms the x input.
y a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647. This forms the y input.

Return Value: This function returns the arc tangent of *y*, divided by *x*, in Q16 format. The value ranges from -2147483648 to 2147483647.

_Q16atanYByXByPI

Description: This function finds the arc tangent of the 32-bit input, *y*, divided by *x*, and then divides by PI (π).

Include: <libq.h>

Prototype: `_Q16 _Q16atanYByXByPI(_Q16 x, _Q16 y);`

Argument: *x* a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647. This forms the x input.
y a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647. This forms the y input.

Return Value: This function returns the arc tangent *y* divided by *x*, divided by PI, in Q16 format. The value ranges from -2147483648 to 2147483647.

16-Bit Language Tools Libraries

_Q16cos

Description: This function finds the cosine of a Q16 value.

Include: `<libq.h>`

Prototype: `_Q16 _Q16cos(_Q16 x);`

Argument: `x` a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647.

Return Value: This function returns the cosine of `x` in Q16 format. The value ranges from -65566 to 65536.

_Q16cosPI

Description: This function finds the cosine of PI (π) times a Q16 value.

Include: `<libq.h>`

Prototype: `_Q16 _Q16cosPI(_Q16 x);`

Argument: `x` a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647.

Return Value: This function returns the cosine of PI times `x` in Q16 format. The value ranges from -65536 to 65536.

_Q16exp

Description: This function finds the exponential value of a Q16 value.

Include: `<libq.h>`

Prototype: `_Q16 _Q16exp(_Q16 x);`

Argument: `x` a fixed point number in Q16 format. The value of this argument ranges from -772244 to 681391.

Return Value: This function returns the exponent value of `x` in Q16 format. The value ranges from 0 to 2147483647.

_Q16log

Description: This function finds the natural log of a Q16 value.

Include: `<libq.h>`

Prototype: `_Q16 _Q16log(_Q16 x);`

Argument: `x` a fixed point number in Q16 format. The value of this argument ranges from 1 to 2147483647.

Return Value: This function returns the natural log of `x` in Q16 format. The value ranges from -726817 to 681391.

_Q16log10

Description: This function finds the log (base 10) of a Q16 value.

Include: `<libq.h>`

Prototype: `_Q16 _Q16log10(_Q16 x);`

Argument: `x` a fixed point number in Q16 format. The value of this argument ranges from 1 to 2147483647.

Return Value: This function returns the log of `x` in Q16 format. The value ranges from -315653 to 295925.

Fixed Point Math Functions

_Q16mac

Description:	This function multiplies the two 32-bit inputs, <i>x</i> and <i>y</i> , and accumulates the product with <i>prod</i> . The function takes care of saturating the result in case of underflow or overflow.
Include:	<libq.h>
Prototype:	<code>_Q16 _Q16mac(_Q16 x, _Q16 y, _Q16 prod);</code>
Argument:	<i>x</i> a fixed point number in Q16 format. The value of this argument ranges from 0 to 2147483647. <i>y</i> a fixed point number in Q16 format. The value of this argument ranges from 0 to 2147483647. <i>prod</i> a fixed point number in Q16 format. The value of this argument ranges from 0 to 2147483647.
Return Value:	This function returns the multiplied and accumulated value <i>prod</i> in Q16 format. The value ranges from 0 to 2147483647.

_Q16macNoSat

Description:	This function multiplies the two 32 bit inputs, <i>x</i> and <i>y</i> and accumulates the product with <i>prod</i> . This function only sets the <code>_Q16macSatFlag</code> variable in case of an overflow or underflow and does not take care of saturation.
Include:	<libq.h>
Prototype:	<code>_Q16 _Q16macNoSat(_Q16 x, _Q16 y, _Q16 prod);</code>
Argument:	<i>x</i> a fixed point number in Q16 format. The value of this argument ranges from 0 to 2147483647. <i>y</i> a fixed point number in Q16 format. The value of this argument ranges from 0 to 2147483647. <i>prod</i> a fixed point number in Q16 format. The value of this argument ranges from 0 to 2147483647.
Return Value:	This function returns the multiplied and accumulated value <i>prod</i> in Q16 format. The value ranges from 0 to 2147483647.

_Q16neg

Description:	This function negates <i>x</i> with saturation. The value is saturated in the case where the input is -2147483648.
Include:	<libq.h>
Prototype:	<code>_Q16 _Q16neg(_Q16 x);</code>
Argument:	<i>x</i> a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647.
Return Value:	This function returns the negated value of <i>x</i> in Q16 format. The value ranges from -2147483648 to 2147483647.

16-Bit Language Tools Libraries

_Q16norm

Description: This function finds the normalized value of a Q16 value.

Include: `<libq.h>`

Prototype: `_Q16 _Q16norm(_Q16 x);`

Argument: `x` a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647.

Return Value: This function returns the square root value of `x` in Q16 format. The value ranges from 1073741824 to 2147483647 for a positive number and -2147483648 to -1073741824 for a negative number.

_Q16power

Description: This function finds the power result, given the base value, `x`, and the power value, `y`.

Include: `<libq.h>`

Prototype: `_Q16 _Q16power(_Q16 x, _Q16 y);`

Argument: `x` a fixed point number in Q16 format. The value of this argument ranges from 0 to 2147483647.
`y` a fixed point number in Q16 format. The value of this argument ranges from 0 to 2147483647.

Return Value: This function returns the value of `x` to the power of `y` in Q16 format. The value ranges from 0 to 2147483647.

_Q16random

Description: This function generates a pseudo random number with a period of 2147483648. The function uses the `_Q16randomSeed` variable as a random seed value.

Include: `<libq.h>`

Prototype: `_Q16 _Q16random(void);`

Argument: None.

Return Value: This function returns the generated random number in Q16 format. The value of this output ranges from -2147483648 to 2147483647.

Remarks: $\text{RndNum}(n) = (\text{RndNum}(n-1) * \text{RAN_MULT}) + \text{RAN_INC}$
SEED VALUE = 21845, RAN_MULT = 1664525 and RAN_INC = 1013904223.

_Q16reciprocal

Description: This function returns the reciprocal of a Q16 value.

Include: `<libq.h>`

Prototype: `_Q16 _Q16reciprocal(_Q16 x);`

Argument: `x` a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647.

Return Value: This function returns the reciprocal of `x` in Q16 format. The value of this output ranges from -2147483648 to 2147483647.

Fixed Point Math Functions

_Q16reciprocalQ15

Description:	This function returns the reciprocal of a Q15 value. Since the input range lies in the -1 to +1 region, the output is always greater than the -1 or +1 region. So, Q16 format is used to represent the output.
Include:	<libq.h>
Prototype:	<code>_Q16 _Q16reciprocalQ15(_Q15 x);</code>
Argument:	<i>x</i> a fixed point number in Q15 format, which ranges from -2^{15} to $2^{15}-1$. The value of this argument ranges from -32768 to 32767.
Return Value:	This function returns the reciprocal of <i>x</i> in Q16 format. The value ranges from -2147483648 to 2147418112.

_Q16reciprocalQ16

Description:	This function returns the reciprocal value of the input.
Include:	<libq.h>
Prototype:	<code>_Q16 _Q16reciprocalQ16(_Q16 x);</code>
Argument:	<i>x</i> a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647.
Return Value:	This function returns the reciprocal of <i>x</i> in Q16 format. The value of this output ranges from -2147483648 to 2147483647.

_Q16shl

Description:	This function shifts the input argument, <i>x</i> , by <i>y</i> number of bits, to the left if <i>y</i> is positive or to the right if <i>y</i> is negative. The function takes care of saturating the result, in case of underflow or overflow.
Include:	<libq.h>
Prototype:	<code>_Q16 _Q16shl(_Q16 x, short y);</code>
Argument:	<i>x</i> a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647. <i>y</i> an integer number, which ranges from -32 to +32.
Return Value:	This function returns the shifted value of <i>x</i> in Q16 format. The value ranges from -2147483648 to 2147483647.

_Q16shlNoSat

Description:	This function shifts the input argument, <i>x</i> , by <i>y</i> number of bits, to the left if <i>y</i> is positive or to the right if <i>y</i> is negative. The function sets the <code>_Q16shlSatFlag</code> variable in case of underflow or overflow, but does not take care of saturation.
Include:	<libq.h>
Prototype:	<code>_Q16 _Q16shlNoSat(_Q16 x, short y);</code>
Argument:	<i>x</i> a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647. <i>y</i> an integer number, which ranges from -32 to +32.
Return Value:	This function returns the shifted value of <i>x</i> in Q16 format. The value ranges from -2147483648 to 2147483647.

16-Bit Language Tools Libraries

_Q16shr

Description:	This function shifts the input argument, <i>x</i> , by <i>y</i> number of bits, to the right if <i>y</i> is positive or to the left if <i>y</i> is negative. The function takes care of saturating the result, in case of underflow or overflow.
Include:	<libq.h>
Prototype:	<code>_Q16 _Q16shr(_Q16 x, short y);</code>
Argument:	<i>x</i> a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647. <i>y</i> an integer number, which ranges from -32 to +32.
Return Value:	This function returns the shifted value of <i>x</i> in Q16 format. The value ranges from -2147483648 to 2147483647.

_Q16shrNoSat

Description:	This function shifts the input argument, <i>x</i> , by <i>y</i> number of bits, to the right if <i>y</i> is positive or to the left if <i>y</i> is negative. The function sets the <code>_Q16shrSatFlag</code> variable in case of underflow or overflow, but does not take care of saturation.
Include:	<libq.h>
Prototype:	<code>_Q16 _Q16shrNoSat(_Q16 x, short y);</code>
Argument:	<i>x</i> a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647. <i>y</i> an integer number, which ranges from -32 to +32.
Return Value:	This function returns the shifted value of <i>x</i> in Q16 format. The value ranges from -2147483648 to 2147483647.

_Q16sin

Description:	This function finds the sine of a Q16 value.
Include:	<libq.h>
Prototype:	<code>_Q16 _Q16sin(_Q16 x);</code>
Argument:	<i>x</i> a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647.
Return Value:	This function returns the sine of <i>x</i> in Q16 format. The value ranges from -65536 to 65536.

_Q16sinPI

Description:	This function finds the sine of PI (π) times a Q16 value.
Include:	<libq.h>
Prototype:	<code>_Q16 _Q16sinPI(_Q16 x);</code>
Argument:	<i>x</i> a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647.
Return Value:	This function returns the sine of PI times <i>x</i> in Q16 format. The value ranges from -65536 to 65536.

Fixed Point Math Functions

_Q16sinSeries

Description:	Generates the sine series with the given normalizing frequency, <i>f</i> , and the given number of samples, <i>num</i> , starting from <i>start</i> . Stores the result in buffer, <i>buf</i> .
Include:	<libq.h>
Prototype:	short _Q16sinSeries(_Q16 f, short start, short num, _Q16 *buf);
Argument:	<i>f</i> a fixed point number in Q16 format, which ranges from 0 to $(2^{31}-1)$. The valid range of values for this argument is from -32768 to 32768. This argument represents the Normalizing frequency. <i>start</i> a fixed point number in Q16 format, which ranges from 0 to $(2^{31}-1)$. The valid range of values for this argument is from 1 to 32767. This argument represents the Starting Sample number in the Sine Series. <i>num</i> a fixed point number in Q16 format, which ranges from 0 to $(2^{31}-1)$. The valid range of values for this argument is from 1 to 32767. This argument represents the Number of Sine Samples the function is called to generate. Note: <i>num</i> should not be more than 16383 for dsPIC and 32767 for PIC devices. <i>buf</i> a pointer to the buffer where the generated sine samples would get copied into.
Return Value:	This function returns <i>num</i> , the number of generated sine samples.

_Q16tan

Description:	This function finds the tangent of a Q16 value.
Include:	<libq.h>
Prototype:	_Q16 _Q16tan(_Q16 x);
Argument:	<i>x</i> a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647.
Return Value:	This function returns the tangent of <i>x</i> in Q16 format. The value ranges from -2147483648 to 2147483647.

_Q16tanPI

Description:	This function finds the tangent of PI (π) times a Q16 value.
Include:	<libq.h>
Prototype:	_Q16 _Q16tanPI(_Q16 x);
Argument:	<i>x</i> a fixed point number in Q16 format. The value of this argument ranges from -2147483648 to 2147483647.
Return Value:	This function returns the tangent of PI times <i>x</i> in Q16 format. The value ranges from -2147483648 to 2147483647.

16-Bit Language Tools Libraries

_Q16ftoi

Description: This function converts a `float` value to a Q16 fixed point fractional value.

Include: `<libq.h>`

Prototype: `_Q16 _Q16ftoi(float f);`

Argument: `f` a floating-point number. The value of this argument ranges from -32768 to 32768.

Return Value: This function returns a Q16 fixed point fractional value. The value ranges from -2147483648 to 2147483647.

_itofQ16

Description: This function converts a Q16 fixed point fractional value to a `float` value.

Include: `<libq.h>`

Prototype: `float _Q16ftoi(_Q16 q);`

Argument: `q` a fixed point number. The value of this argument ranges from -2147483648 to 2147483647.

Return Value: This function returns a floating-point value. The value ranges from -32768 to 32768.
