MRB MATH LIBRARY

DOCUMENTATION

This library is intended for use in lower-cost, slower microcontrollers not dedicated to mathematical calculations (e.g. Arduino, Cortex M0 and M4, ESP, Raspberry Pi) to speed up mathematical operations. It can also be used in DSP to reduce cycle time, wherever cycle time is very critical, but the accuracy of calculations is not so critical. The MRB\_MATH library will also significantly speed up the execution time of math calculations on microcontrollers without FPU or without hardware support for the default math.h library. An additional function not included in the math.h library is the function for calculating the RMS of a signal. This function is available in three variants: normal, fast and rapid. The normal version uses functions from the math.h library. The fast one uses functions from the MRB\_MATH library, and the rapid one does not use division or root operations at all (but it is adektwat only for signals very close to an ideal sine wave signal).

In summary, this library allows faster calculations at the expense of lower precision and higher memory usage

The documentation describes the functions contained in the MRB\_MATH library and compares them with their counterparts from the math.h library. Execution times and precision are compared. The results are summarized in graphs or tables. The possibility to parameterize each function is also described - the MRB\_MATH library makes it possible to increase the precision of a function at the cost of additional cycles or FLASH/RAM memory usage.

Functions included in the library

* **sin\_f**(float x) sinus from argument x
* **cos\_f**(float x) cosinus from argument x
* **fast\_invsqrt**(float x) fast inverse root square from argument x
* **fast\_sqrt**(float x) fast root square from argument x
* **RMS**(float x) root mean square function (with normal sqrt function)
* **fast\_RMS**(float x) fast root mean square function (with fast sqrt function)
* **rapid\_RMS**(float x) rapid root mean square function (approach without sqrt)

Execution speed overview

Table 1 Comparison of function execution time in terms of cycles on STM32 microcontroller with Cortex M-7 core

|  |  |  |
| --- | --- | --- |
| **Function / Library** | **<math.h>** | **“MRB\_MATH\_LIB.h”** |
| sinus | 762 | 90 |
| cosinus | 762 | 92 |
| sqrt | 674 | 28 |

Library created by Maciej Brzycki ------ February 2024

# Trigonometric functions

The sine and cosine functions in the MRB\_MATH library are based on the look up table, which will be stored in the microcontroller's flash memory when the program is uploaded. While the program is running, it permanently occupies space in ram memory. User can define the size of memory usage with *LUTSIZE* parameter. When equal to 1 - look up table will take 4kB of memory (1000 float values), when 2 – 8kB and when 4 – 16kB.

**Important note**: the range of the function's arguments is not as large as that of the sin function from the math.h library. The argument of the sin\_f function must be from the range -2π to 4π.

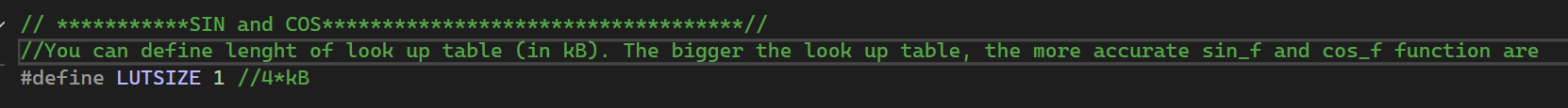


Figure 1 LUT size choice

A screenshot of a computer

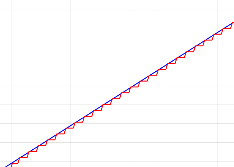
Description automatically generated

Figure 2 LUT view - beginning of the declaration





Figure 3 Use of trigonometric functions

A graph of a function

Description automatically generated

Figure 4 math.h sin(x) function and MRB\_MATH\_LIB.h sin\_f(x) comparison (LUTSIZE = 1)

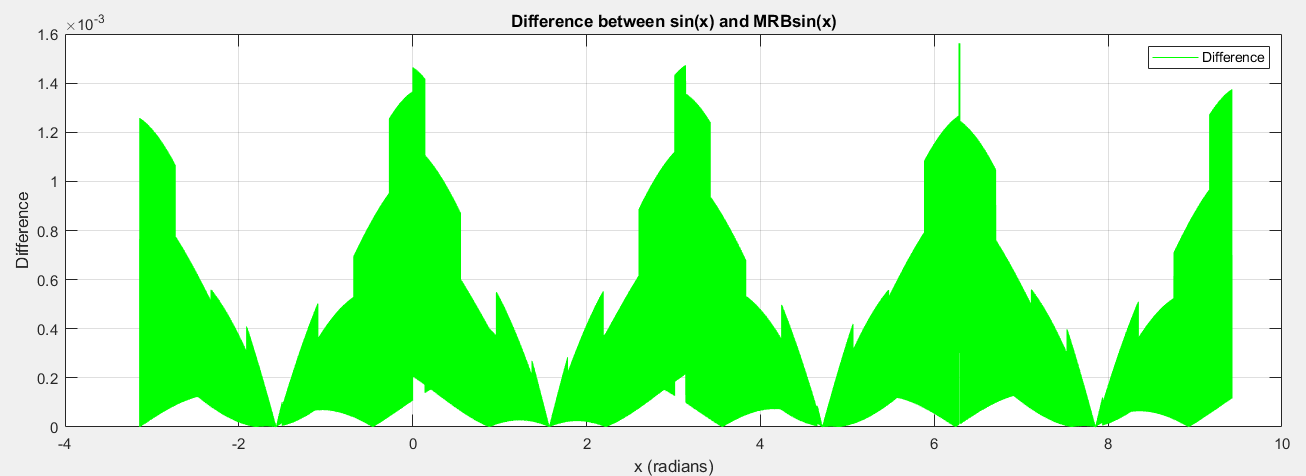


Figure 5 Absolute error between sin and sin\_f functions (LUTSIZE = 1)

Table 2 Mean absolute difference between sin and sin\_f

|  |  |  |  |
| --- | --- | --- | --- |
| **LUTSIZE(n\*4kB)** | **1** | **2** | **4** |
| **Mean Absolute error** | 0.000326 | 0.000163 |  |