

# FRA222 Microcontroller Interface

---

09

ARM CMSIS DSP LIBRARY



# DSP – Digital signal Processing library

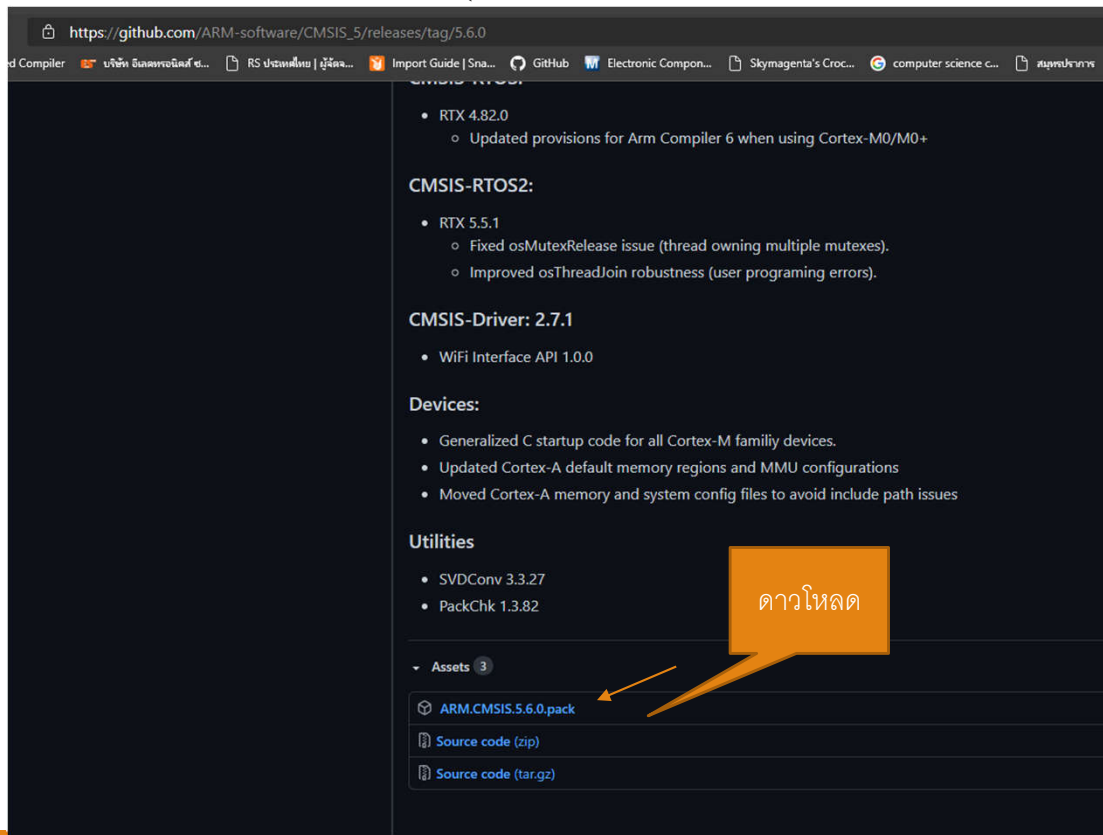
---

- ช่วยในการคำนวณ ข้อมูลสัญญาณ ในรูปแบบ ดิจิตอล เช่น lowpass filter FFT PID etc.
- เสริมฟังก์ชันการทำงานทางคณิตศาสตร์ ให้กับระบบ
- เพิ่มความเร็วการคำนวณทางเมทริก(บางส่วน)

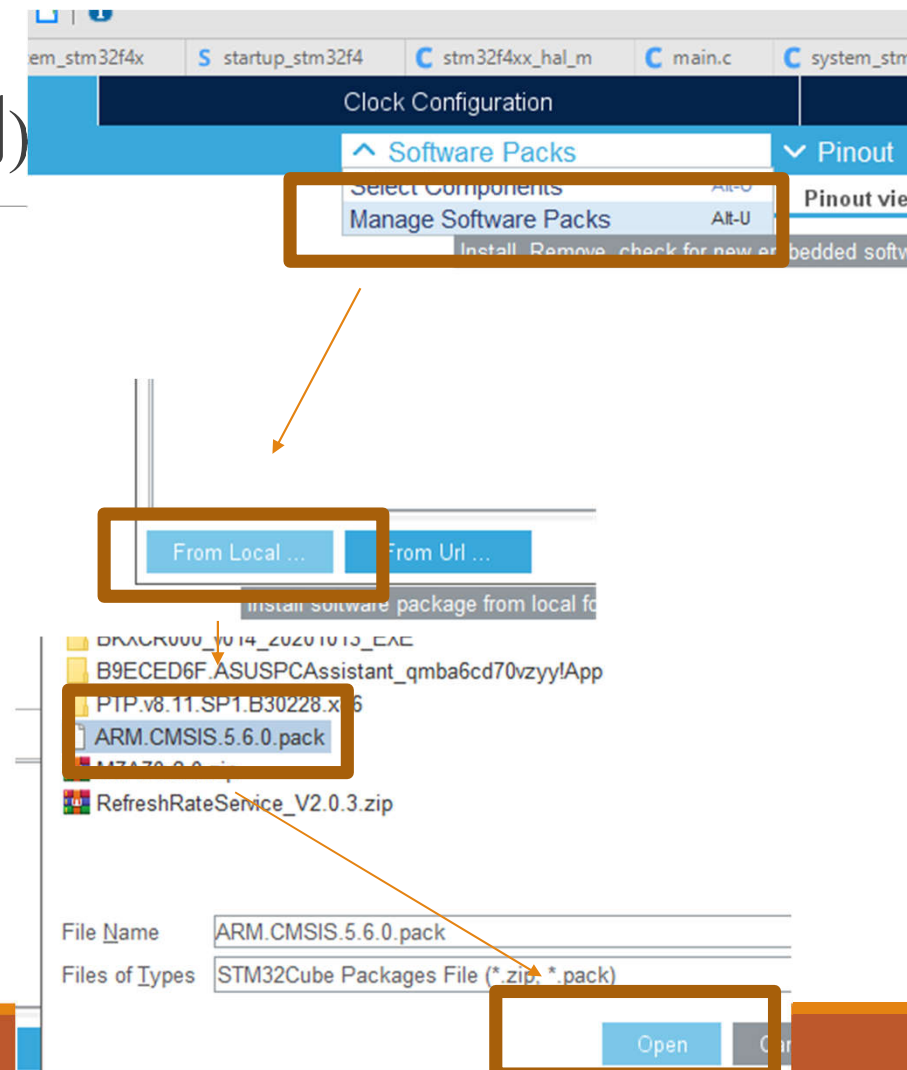
การติดตั้งมีสองวิธี

1. ติดตั้งโดยในระบบอัตโนมัติ สำหรับ **lib version** เก่า
2. เพิ่ม**source** ด้วยมือ สำหรับ **lib version** ใหม่

# การติดตั้ง (version 5.6.0 ลงไป)



[https://github.com/ARM-software/CMSIS\\_5](https://github.com/ARM-software/CMSIS_5)



# การติดตั้ง

---

Releases information was last refreshed less than one hour ago.

STM32Cube MCU Packages	STMicroelectronics	ARM	RoweBots	wolfSSL
	Status	Description	Available Version	
▼		CMSIS		
■		CMSIS (Cortex Microcontroller Software Interface Standard)	5.6.0	

[https://github.com/ARM-software/CMSIS\\_5](https://github.com/ARM-software/CMSIS_5)

# การติดตั้ง

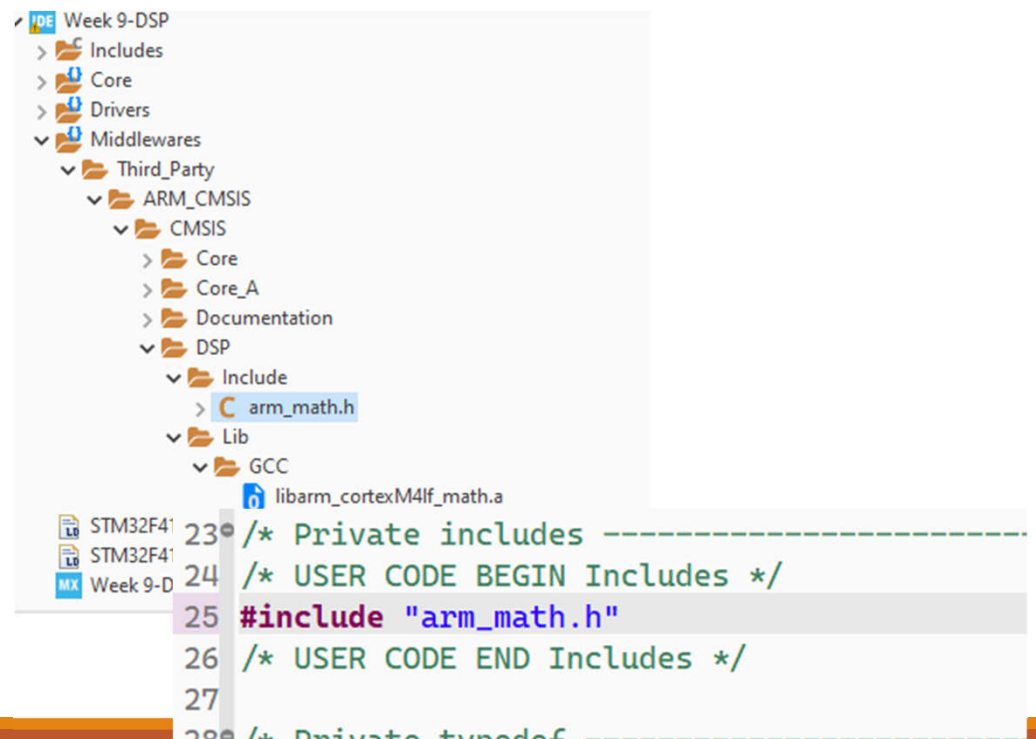
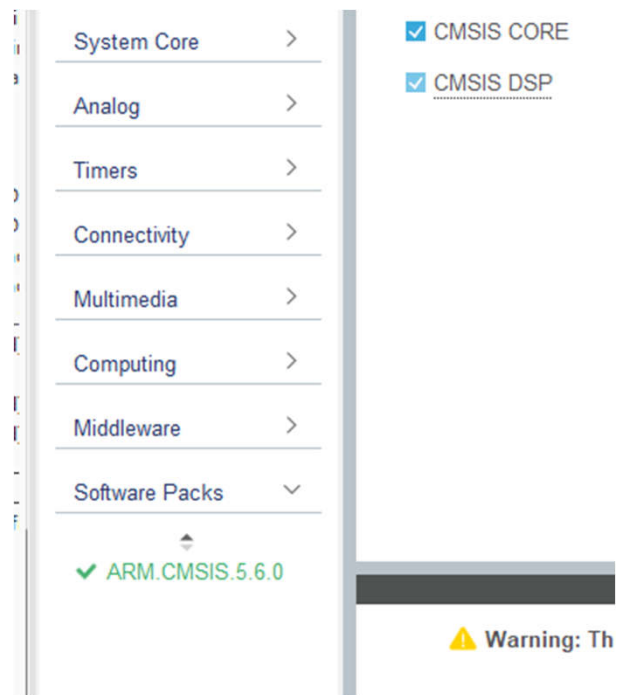
Software Packs Component Selector

Packs

Pack / Bundle / Component	Status	Version	Selection
ARM.CMSIS	✓	5.6.0	
Exposed APIs			
CMSIS CORE	✓	5.3.0	<input checked="" type="checkbox"/>
Device Startup			Not selected ▾
Device IRQ Controller / GIC	⚙	1.0.1	<input type="checkbox"/>
Device OS Tick / Private Timer	⚙	1.0.2	<input type="checkbox"/>
Device OS Tick / Generic Physical Timer	⚙	1.0.1	<input type="checkbox"/>
CMSIS DSP	✓	1.7.0	Library ▾
CMSIS NN Lib		1.2.0	Not selected
CMSIS RTOS / Keil RTX			Library
CMSIS RTOS / Keil RTX5		5.5.1	Source
CMSIS RTOS2 / Keil RTX5			Not selected ▾
CMSIS Driver USART / Custom		2.3.0	<input type="checkbox"/>
CMSIS Driver SPI / Custom		2.2.0	<input type="checkbox"/>
CMSIS Driver SAL / Custom		1.1.0	<input type="checkbox"/>

[https://github.com/ARM-software/CMSIS\\_5](https://github.com/ARM-software/CMSIS_5)

# การติดตั้ง



# การติดตั้ง (CMSIS 5.7.0++ \ DSP 1.14.4)

<https://github.com/ARM-software/CMSIS-DSP/releases/tag/v1.14.4>

Releases / v1.14.4

## CMSIS-DSP 1.14.4 Latest

christophe0606 released this last week · 7 commits to main since this release · v1.14.4 · 3a04f81

### What's Changed

- Missing header added to `arm_math.h`
- Documentation improvements
- Improvements to the generic nodes of the compute graph

### Assets

- ARM.CMSIS-DSP1.14.4.pack 7.58 MB
- Source code (zip)**
- Source code (tar.gz)

1 person reacted

..	File folder
Testing	File folder
Source	File folder
Scripts	File folder
PythonWrapper	File folder
PrivateInclude	File folder
Include	File folder
Examples	File folder
Doxygen	File folder
Documentation	File folder
ComputeLibrary	File folder
ComputeGraph	File folder
cmsisdsp	File folder
.github	File folder
setup.py	2,710 JetBrains PyCh
README.md	5,466 MD File
PythonWrapper_RE	1,534 MD File
pyproject.toml	102 TOML File
MANIFEST.in	149 IN File
LICENSE.txt	3,948 Text Document
cmsisdspconfig.py	1,438 JetBrains PyCh
CMakeLists.txt	317 Text Document
ARM.CMSIS-DSP.p	1,770 PDSC File
.gitignore	318 Text Document
.gitconfig	66 GITCONFIG File

Select all

Ctrl+A

Copy files to clipboard

Ctrl+C

Copy full names to clipboard

File folder

Add files to archive

Alt+A

Extract to a specified folder

Alt+E

Extract without confirmation

Alt+W

Delete files

Del

View file

Alt+V

Create a new folder

Rename

View as

Sort by

DSP-1.14.4

Name	Date modified	Type	Size
ComputeLibrary	10-Mar-23 20:55	File folder	
Include	10-Mar-23 20:55	File folder	
PrivateInclude	10-Mar-23 20:55	File folder	
Source	10-Mar-23 20:55	File folder	

Pinout & Configuration

Clock Configuration

IDE DSPTEST

New

Go Into

Open in New Window

Show In Alt+Shift+W

Copy Ctrl+C

Paste Ctrl+V

Delete Delete

Source

Project...

File

File from Template

Folder

Create a new folder resource

Header File

Source File

Source Folder

IDE DSPTESTPROJECT

Includes

Core

Drivers

DSP

DSPTESTPROJECT.ioc

STM32F411RETX\_FLASH.Id

STM32F411RETX\_RAM.Id



Includes

Core

Drivers

DSP

DSPTESTPROJECT.ioc

STM32F411RETX\_FLASH.Id

STM32F411RETX\_RAM.Id

GPIO (in Program) [W1GPI

GPIO1 (in GPIO) [GPIO mair

InterruptAndDMA

testDSP2

testDSP2023

testdsp3

testDSP4

testFreeRTOS

Week10-UART (in 10) [Week

week6ioc (in ICOC) [Week

project\_3

project\_5

project\_6

project\_7

project\_8

project\_9

project\_1

Name

ComputeLibrary

Include

PrivateInclude

Source

File and Folder Operation

Select how files and folders should be imported into the project:

☐ Copy files and folders

☐ Link to files and folders

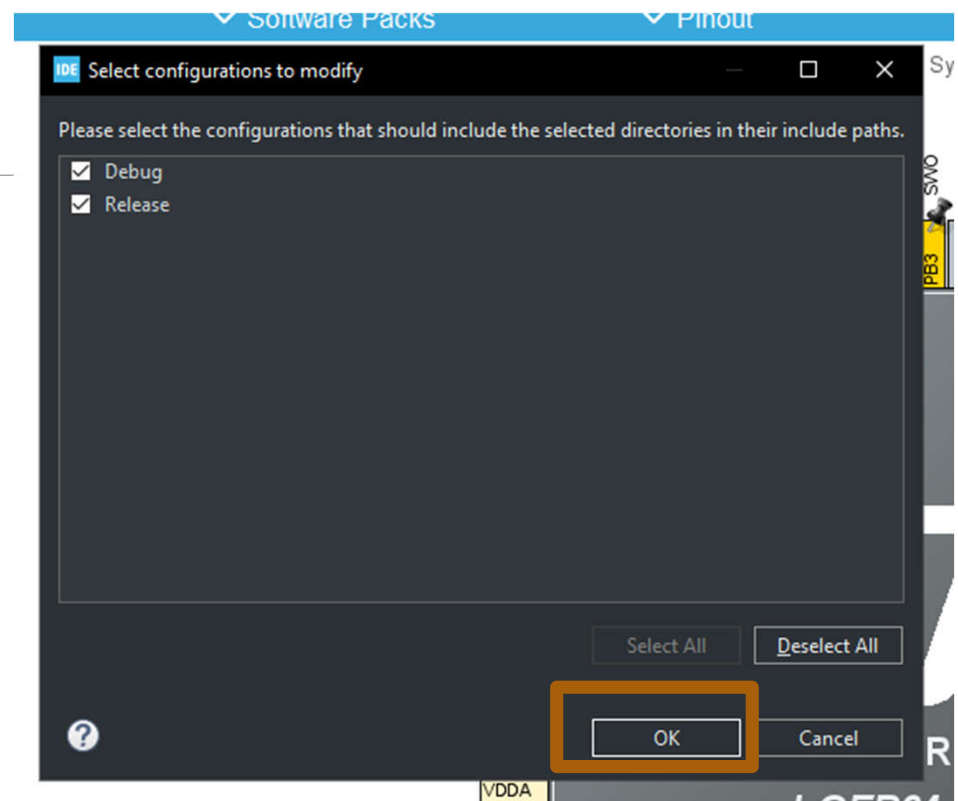
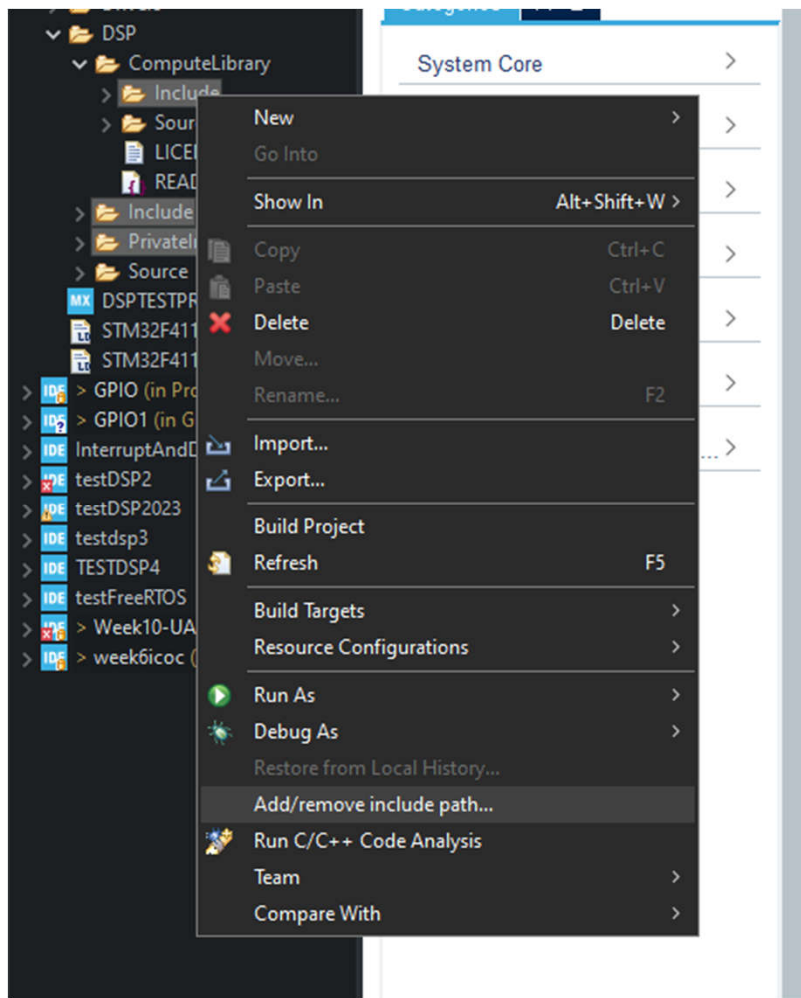
☐ Link to files and recreate folder structure with virtual folders

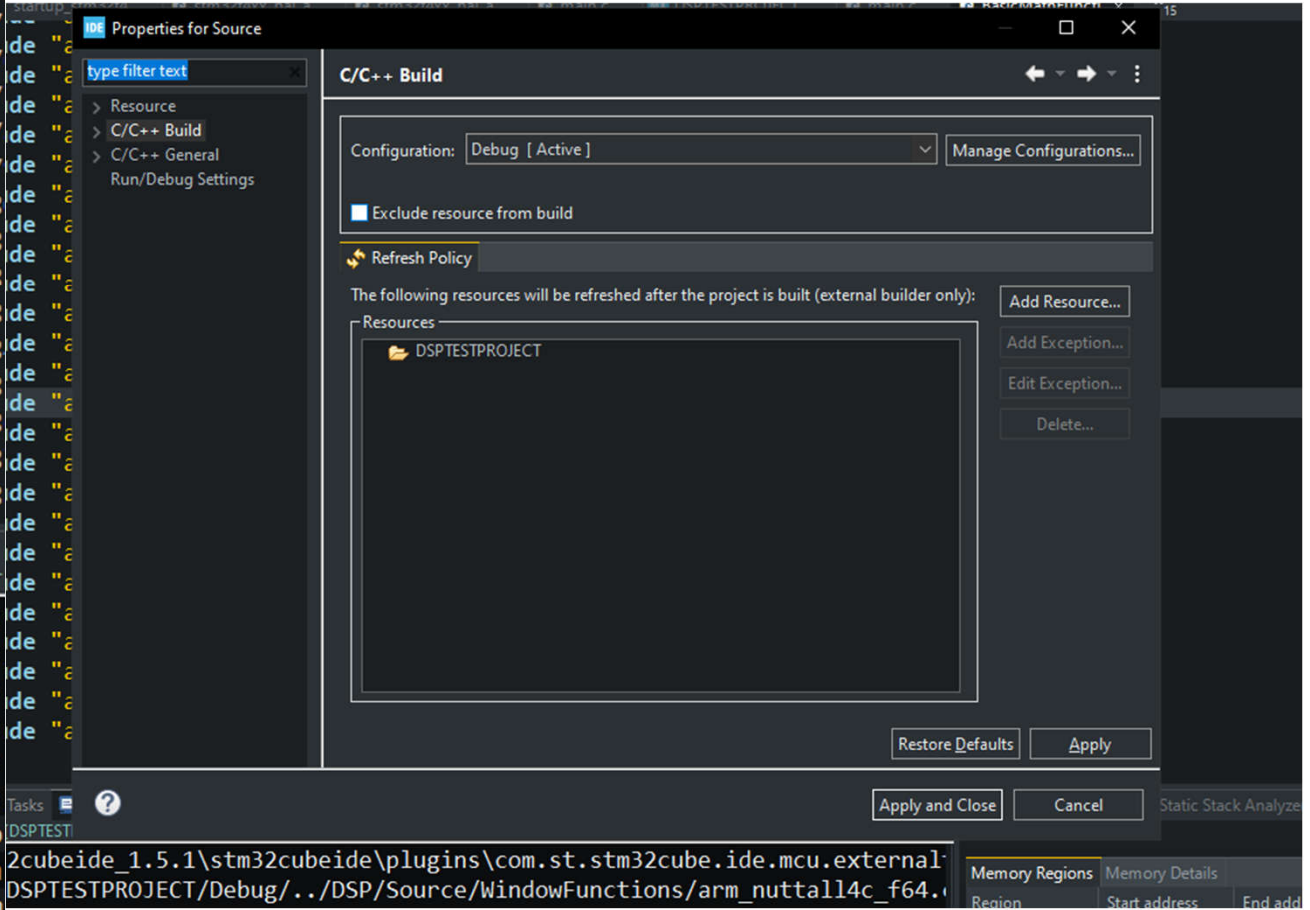
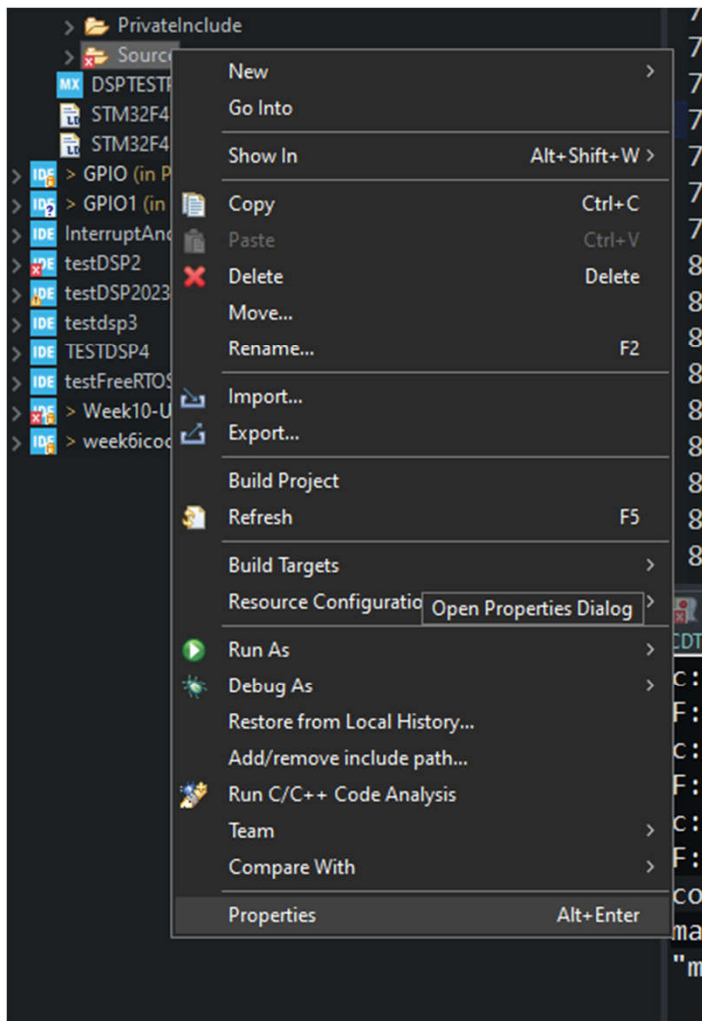
☒ Create link locations relative to: PROJECT\_LOC

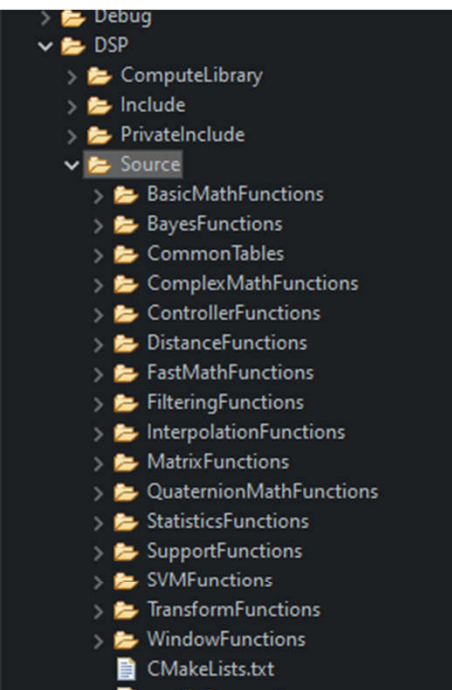
[Configure Drag and Drop Settings...](#)

OK Cancel

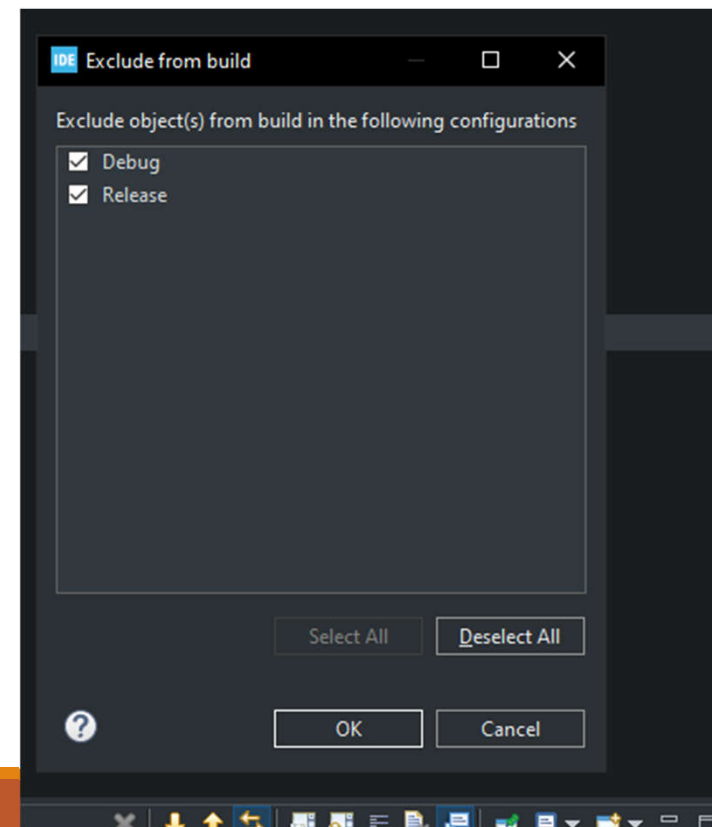
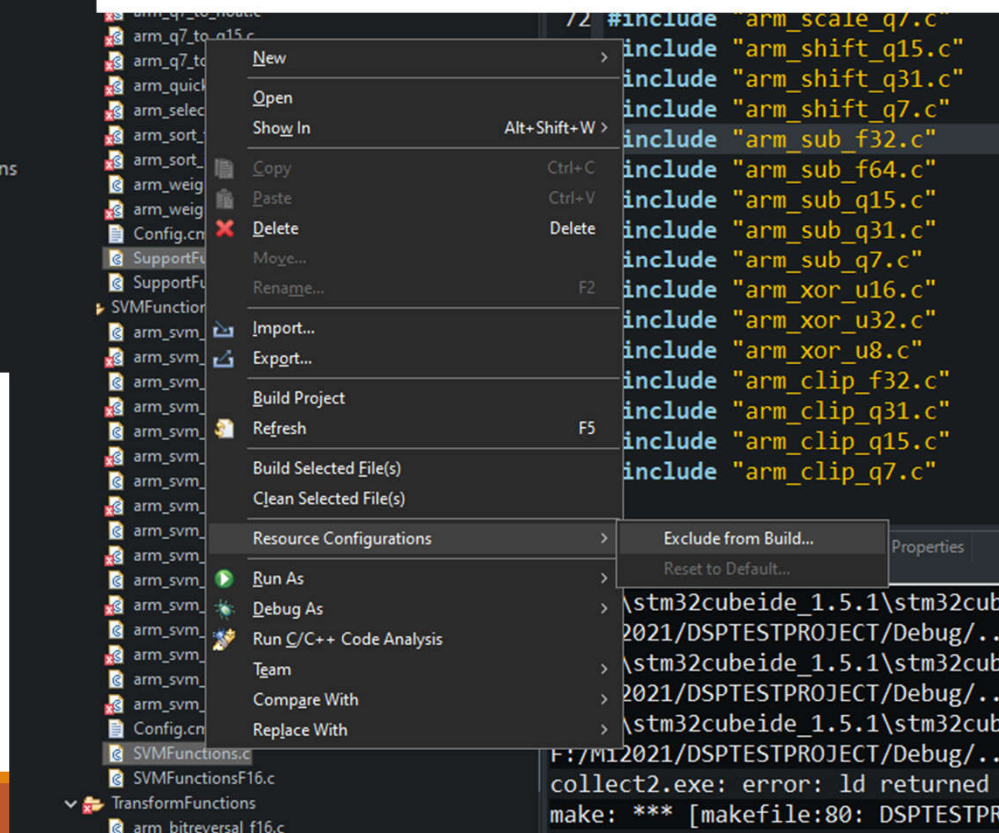








ใน folder “Source” ในแต่ละ sub folder ให้เลือก .c ไฟล์ที่ชื่อตรงกับ โฟลเดอร์ และทำการ exclude ออกจาก project



```
17  */
18  /* USER CODE END Header */
19  /* Includes -----*/
20  #include "main.h"
21
22  /* Private includes -----*/
23  /* USER CODE BEGIN Includes */
24  #include "arm_math.h"
25  /* USER CODE END Includes */
26
```

```
while (1)
{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
    arm_sin_f32(0);
}
```

# Variable

```
arm_rfft_instance_q31 : struct
arm_status : enum
float32_t : float
float64_t : double
q15_t : int16_t
q31_t : int32_t
q63_t : int64_t
q7_t : int8_t
arm_abs_f32(const float32_t*, float32_t*, uint32_t) : void
arm_abs_q15(const q15_t*, q15_t*, uint32_t) : void
arm_abs_q31(const q31_t*, q31_t*, uint32_t) : void
arm_abs_q7(const q7_t*, q7_t*, uint32_t) : void
arm_add_f32(const float32_t*, const float32_t*, float32_t*, uint32_t) : void
arm_add_q15(const q15_t*, const q15_t*, q15_t*, uint32_t) : void
arm_add_q31(const q31_t*, const q31_t*, q31_t*, uint32_t) : void
arm_add_q7(const q7_t*, const q7_t*, q7_t*, uint32_t) : void
```

1. fxx – f32 f64 ➔ float32\_t, float64\_t = Floating point

- Just a totally normal Floating point

2. qxx – q7, q15, q31, q63 ➔ Fixed point

- not a normal integer

# FIXED Point ?

---

0000 0000 0000 0000 0000 0000 0000 0000      32bit

float32\_t      1Sign 8exponent      23fraction

0000 0000 0000 0000 0000 0000 0000 0000

2<sup>23</sup> = 8,388,608 step

Number of decimal digits : 6-7 digits , range +- 3.4028234664 × 10<sup>38</sup>

$$\text{value} = (-1)^{\text{sign}} \times 2^{(E-127)} \times \left( 1 + \sum_{i=1}^{23} b_{23-i} 2^{-i} \right).$$

1Sign      31decimal

int32\_t      0000 0000 0000 0000 0000 0000 0000 0000

2<sup>31</sup> = 2,147,483,648 step

Number of decimal digits : 9 – 10 digits range -2,147,483,648 to 2,147,483,647

# FIXED Point ?

---

Q Format Qm.n

*Range* =  $-(2^{m-1})$  to  $2^{m-1} - 2^{-n}$

*Resolution* :  $2^{-n}$

$$-(2^{32-1}) = -2147483648$$

int32\_t (Q32.0 Format)

$$2^{32-1} - 2^{-0} = 2147483647$$

$$-2^{1-1} = -1$$

q31\_t (Q1.31 Format)

$$2^{1-1} - 2^{-31} =$$
$$0.999999999953433871269226$$
$$07421875$$

in Lib

q31\_t = Q1.31 ,

q15\_t = Q1.15 ,

q64\_t = Q1.64

# why? use Qformat in DSP

---

- Better Number of decimal digits , in same size of storage float32 (6-7 digit) | q31(9-10 digit)
- Faster Calculation\*
- Easy to convert in microcontroller
- Cost effective in Hardware design

\*When use with optimization and in right situation





# แล้ว...เราต้องใช้Q format ไหม? ในตอนนี้

---

## 1. float32\_t vs q31\_t – อันไหนก็ได้

- 1. STM32F4 มี FPU ระดับ single point ซึ่งช่วยเร่งความเร็วในการคำนวณ Float อยู่แล้ว รวมๆ performance ไม่ต่างกันมาก
- 2. cost ที่เพิ่มมาจากการใช้ float ส่วนมากมาจากการแปลง ไปๆ กลับๆ ระหว่าง float กับ int ซึ่งเกิดบ่อย เวลา ใช้กับ peripheral ต่างๆ เช่น ADC TIM
- 3. q31 เป็น implementation เพราะฉะนั้น คำนวณโดยใช้การ +-/ แบบปกติไม่ได้ ต้องใช้ฟังก์ชันเฉพาะ (ซึ่งเป็นเหตุผลให้ ข้อ1 ที่รวมๆแล้ว ใช้แบบไหนก็ได้)

## 2. float64\_t vs q63\_t – q63\_t

1. STM32F4 มี FPU ระดับ single point มั่นแรง double ไม่ได้มาก ทำให้เกิดความแตกต่างกันอย่างมีนัยยะ

อย่างไรก็ตาม การ implement Qformat ค่อนข้างต้องใช้ความเข้าใจในทางคณิตศาสตร์และคอมพิวเตอร์สูง (เพราะทุกอย่างคำนวณใน range +-1) ต้อง scale ตัวแปรและค่าคงที่ที่ถูกต้องด้วยจึงจะใช้งานได้

# DSP Library

<https://www.keil.com/pack/doc/CMSIS/DSP/html/index.html>

## ▼ Basic Math Functions

- Vector Absolute Value
- Vector Addition
- Vector bitwise AND
- Vector Dot Product
- Vector Multiplication
- Vector Negate
- Vector bitwise NOT
- Vector Offset
- Vector bitwise inclusive OR
- Vector Scale
- Vector Shift
- Vector Subtraction
- Vector bitwise exclusive OR

## ▼ Fast Math Functions

- Square Root
- Cosine
- Sine

## ▼ Complex Math Functions

- Complex Conjugate
- Complex Dot Product
- Complex Magnitude
- Complex Magnitude Squared
- Complex-by-Complex Multiplication
- Complex-by-Real Multiplication

## ▼ Filtering Functions

- High Precision Q31 Biquad Cascade Filter
- Biquad Cascade IIR Filters Using Direct Form I Structure
- Biquad Cascade IIR Filters Using a Direct Form II Transposed Structure
- Convolution

## Partial Convolution

### Correlation

- Finite Impulse Response (FIR) Decimator
- Finite Impulse Response (FIR) Filters
- Finite Impulse Response (FIR) Lattice Filters
- Finite Impulse Response (FIR) Sparse Filters
- Infinite Impulse Response (IIR) Lattice Filters
- Least Mean Square (LMS) Filters
- Normalized LMS Filters
- Finite Impulse Response (FIR) Interpolator

## ▼ Matrix Functions

- Matrix Addition
- Complex Matrix Multiplication
- Matrix Initialization
- Matrix Inverse
- Matrix Multiplication
- Matrix Scale
- Matrix Subtraction
- Matrix Transpose

## ▼ Transform Functions

- Complex FFT Functions
- DCT Type IV Functions
- Real FFT Functions

## ▼ Controller Functions

- PID Motor Control
- Vector Clarke Transform
- Vector Inverse Clarke Transform
- Vector Park Transform

## Vector Inverse Park transform

### Sine Cosine

## ▼ Statistics Functions

- Maximum
- Mean
- Minimum
- Power
- Root mean square (RMS)
- Standard deviation
- Variance

## ▼ Support Functions

- Vector sorting algorithms
- Vector Copy
- Vector Fill
- Convert 32-bit floating point value
- Convert 16-bit Integer value
- Convert 32-bit Integer value
- Convert 8-bit Integer value
- Cubic Spline Interpolation

## ▼ Interpolation Functions

- Linear Interpolation
- Bilinear Interpolation

# vector (basic math function)

---

## Content

Vector Absolute Value

Vector Addition

Vector bitwise AND

Vector Dot Product

Vector Multiplication

Vector Negate

Vector bitwise NOT

Vector Offset

Vector bitwise inclusive OR

Vector Scale

Vector Shift

Vector Subtraction

Vector bitwise exclusive OR

# Fast math function

---

square root

sine

cosine

# matrix

---

```
float32_t A_MAT_BUFF[8]={0};  
arm_matrix_instance_f32 A;  
arm_mat_init_f32(&A, 2, 4, A_MAT_BUFF);
```

```
//NOW A is matrix size 2*4
```

and more

---