M4 DSP Linear Interpolation

Example Code Introduction for 32-bit NuMicro® Family

Information

Application	This sample code uses the M4 DSP to calculate linear interpolation
BSP Version	M480 Series BSP CMSIS V3.04.000
Hardware	NuMaker-PFM-M487 Ver 3.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

1 Function Description

1.1 Introduction

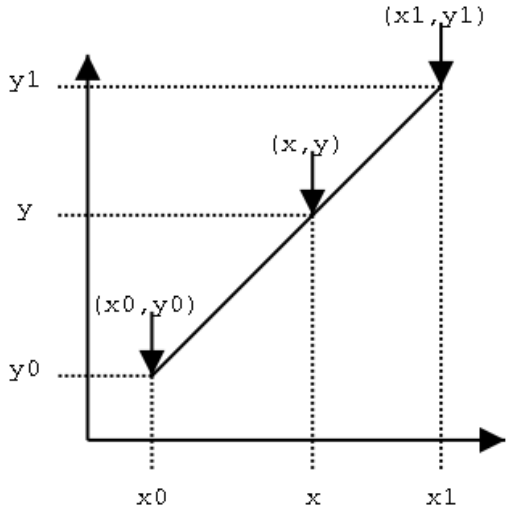
This sample code uses the CMSIS DSP library to calculate Linear Interpolation. User can use these functions to implement mathematical equations. The sample code compares the difference between using the DSP calculation time and calculates the efficiency improvement ratio.

1.2 Principle

Assume there are two coordinates *(x0, y0)* and *(x1, y1)*, and we want to know a value of a (x,y) on the line. According to the image below, the formula is

$$\frac{y - y0}{x - x0} = \frac{y1 - y0}{x1 - x0}$$

User can input value of x to calculate value of y by linear interpolation.



Use CMSIS DSP library to calculate linear interpolation first we need to define number of samples of x0, initial value x0, x1-x0, and y data. The program is set as shown in the following table. After setting the floating-point interpolation structure, input the x-value and then we can calculate y value by linear interpolation.

arm_linear_interp_instance_f32 S = {uint32_t nValues, float32_t x0, float32_t xSpacing, float32_t *pYData}		
Parameters：	nValues	[in] x number of samples
	x0	[in] x initial value x0
	xSpacing	[in] x1-x0
	*pYData	[in] y data
Returns：		none

arm_linear_interp_f32(arm_linear_interp_instance_f32 *S, float32_t x)		
Parameters：	*S	[in, out] S is an instance of the floating-point Linear Interpolation structure
	x	[in] x input sample to process
Returns：		y processed output sample.

1.3 Demo Result

```
+----------------------------------------+
|      DSP Interpolation Sample Code     |
+----------------------------------------+

Calculating time with DSP instruction is 0.004250 ms

Calculating time without DSP instruction is 0.004833 ms

Efficiency increase rate is 1.14
```

2 Code Description

Use CMSIS DSP library to calculate Interpolation：

```
/* Initial Interpolation(x number of samples, x initial value, x value interval, y data)*/
arm_linear_interp_instance_f32 S = {2, 0, 10, (float32_t *)&arm_linear_interep_table[0]};

/* Calculate Linear interpolation with DSP, Calculation results will return to
testLinIntOutput */
for(i=0; i<TEST_LENGTH_SAMPLES; i++) {
    testLinIntOutput[i] = arm_linear_interp_f32(&S, testInputSin_f32[i]);
} ;
```

And calculation Interpolation without DSP：

```
/* Calculate Linear Interpolation */
float interp(int n, float x1, float xSpacing, float x)
{
    float y, y0, y1, x0;                    /* Nearest output values */
    int32_t i;                              /* Index variable */

    /* Calculation of index */
    i = (x - x1) / xSpacing;
    if(i < 0) {
      /* Initialize output for below specified range as least output value of table */
      y = arm_linear_interep_table[0];
    }
    else if(i >= n) {
      /* Iniatilize output for above specified range as last output value of table */
      y = arm_linear_interep_table[n - 1];
    }
    else {
      /* Calculation of nearest input values */
      x0 = x1 + i * xSpacing;
      x1 = x1 + (i + 1) * xSpacing;
      /* Read of nearest output values */
      y0 = arm_linear_interep_table[i];
      y1 = arm_linear_interep_table[i + 1];
      /* Calculation of output */
      y = y0 + (x - x0) * ((y1 - y0) / (x1 - x0));
    }
    /* returns output value */
```

```
    return (y);
}
```

Change the counter to time, where the timer clock source is HXT 12MHz：

```
    /* Calculate the time, timer clock source is 12M, unit is ms */
    DSPCalTime = (DSPCalTime/12000000) * 1000;
    CalTime = (CalTime/12000000)* 1000;
```

3 Software and Hardware Environment

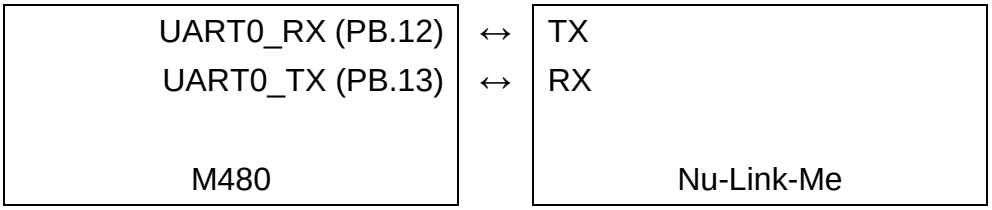
Software environment

- BSP version
  - M480 Series BSP CMSIS V3.04.000
- IDE version
  - Keil uVersion 5.26

Hardware environment

- Circuit components
  - NuMaker-IoT-M487 or other M480 Development Board
- Diagram

M480's UART0_RX (PB.12) and UART0_TX (PB.13) are connected to Nu-Link Me to print the message. Set COM port and Baud. The number of the COM Port can be found in the device manager "NuBridge Virtual Com Port (COMX)" and Baud is set to 115200.



4 Directory Information

- EC_M480_Convolution_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 How to Execute Example Code

1. Browsing into sample code folder by Directory Information (section 4) and double click DSP_Convolution.uvproj
2. Enter Keil compile mode
   a. Build
   b. Download
   c. Start/Stop debug session
3. Enter debug mode
   a. Run

6 Revision History

Date	Revision	Description
Jun. 25, 2019	1.00	1. Initially issued.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.