HW7 MbedOS-DSP

data of Author

HomeWork:HW7
Name:邱士倫
ID:B09901036

githubURL: https://github.com/allenCHIUtw/Embed/tree/main/HW7-Mbed-DSP-programming

Experiment

(1) Your DSP program can implement a low pass filter, like FIR example, or other algorithms, such as FFT, etc.
(2) Before testing real data, you should test the program using a known signal to show the correctness
(3)Review question:\What is the Range of Representable Values in the ARM fixed point q15 (Q1.15) format?
Submission: The github URL and a report about your approach and discussions.

實驗流程

第一題

1.我實作的DSP 是模擬FIR 低頻率波器，因此可以濾除高頻波讓訊號曲線的雜訊降低。
以下是初始化設定

```
float sensor_value = 0;
  int16_t pDataXYZ[3] = {0};
  float pGyroDataXYZ[3] = {0};

  uint32_t i;
  uint32_t mem =0;
  arm_fir_instance_f32 S;
  arm_status arm_status;
  float32_t *input_buff,*output_buff;
  input_buff = new float32_t[sample_size];  //320
  output_buff =new float32_t[sample_size]; //320
  float32_t *input_ptr = &input_buff[0];
  float32_t *output_ptr = &output_buff[0];

   BSP_TSENSOR_Init();
   BSP_HSENSOR_Init();
   BSP_PSENSOR_Init();

   BSP_MAGNETO_Init();
   BSP_GYRO_Init();
   BSP_ACCELERO_Init();

   arm_fir_instance_f32 FIR_insrance;
```

```
    float32_t data_x =0;
    float32_t data_y= 0;
    float32_t data_z =0;
    arm_fir_init_f32(&FIR_insrance, 29, (float32_t *)&firCoeffs32[0],
&firStateF32[0], block_size);
```

2.訊號原來自三軸加速器，為了計算納入xyz三軸的數據做平方根號

```
for(i=0; i<sample_size; i++)  // input collecting data
        {
              BSP_ACCELERO_AccGetXYZ(pDataXYZ);
              data_x= float32_t(pDataXYZ[0]);
              data_y= float32_t(pDataXYZ[1]);
              data_z= float32_t(pDataXYZ[2]);
              float32_t temp = sqrt(data_z*data_z +data_y*data_y +
data_x*data_x);
              input_buff[i] =temp;

        }
```

作為參數設定的值為

```
const float32_t firCoeffs32[29] = {
  -0.0018225230f, -0.0015879294f, +0.0000000000f, +0.0036977508f, +0.0080754303f,
+0.0085302217f, -0.0000000000f, -0.0173976984f,
  -0.0341458607f, -0.0333591565f, +0.0000000000f, +0.0676308395f, +0.1522061835f,
+0.2229246956f, +0.2504960933f, +0.2229246956f,
  +0.1522061835f, +0.0676308395f, +0.0000000000f, -0.0333591565f, -0.0341458607f,
-0.0173976984f, -0.0000000000f, +0.0085302217f,
  +0.0080754303f, +0.0036977508f, +0.0000000000f, -0.0015879294f, -0.0018225230f
};
```

3. FIR處理格式我採用arm_fir_f32 分成八份資 料進行處理

```
 for(i=0; i<blockNumber; i++)
        {
             arm_fir_f32( &FIR_insrance,input_ptr+(i*block_size),output_ptr+
(i*block_size),block_size);

             //printf("X_MAG : %f, Y_MAG: %f, Z_MAG: %f\n",result);
        }
```

4.做出來的結果為:

- 處理前 
- 處理後 

第二題

1.因為原裝的 CMSIS-DSP 無法執行以下程式，因此重新找到 data.c 其中的 float32_t testInput_f32_1kHz_15kHz 以及float32_t refOutput

2.做出來的結果為:

- testInput_f32_1kHz_15kHz  before DSP
- refOutput  After DSP
- TestOutput  After DSP 結論:這一個FIR filter 是有效的

review Question

ARM fixed-point Q1.15 格式是一種用於固定點數字的表示方式
*「1」表示用於整數部分的位數為 1 位。這一位還包括了有符號數字的符號位，意味著數字可以是正數或負數。

- 「15」表示有 15 位用於小數部分。

根據這種格式，可表示值的範圍可以如下計算：

- 最大正值發生在整數位為 0（表示正數）且所有小數位為 1 的情況下。這給出了二進制中的 0.111111111111111，大約等於十進制中的 0.999969482421875。
- 最小負值發生在整數位為 1（表示負數）且所有小數位為 0 的情況下。這給出了二進制中的 1.000000000000000，當用二補數表示法解釋時，等於十進制中的 -1。

因此，ARM 固定點 Q1.15 格式的可表示值範圍是從 -1 到大約 0.999969482421875。

實驗感想

這次比較困難的部分在於讓範例程式跑動，將 external 的檔案替換為
https://github.com/ARM-software/CMSIS-DSP/blob/main/Examples/ARM/arm_fir_example/arm_fir_data.c
中的資料都沒辦法如實解決
至於選擇Dir的原因是因為數值比較好填寫操作相對簡單。

參考資料

https://github.com/ARM-software/CMSIS-DSP/tree/main
https://blog.csdn.net/lyd0813/article/details/102575213