Display CCD data in waveform (purchased separately)

1. Learning Objectives

After adding the dynamic threshold algorithm, we get the threshold and median. This time, we will draw the waveform of the 128 data obtained from the camera on the LCD screen (purchased separately) and verify whether the data processed by the algorithm is correct.

2. Introduction of Algorithms

What is an algorithm?

Baidu Encyclopedia says: Algorithm refers to an accurate and complete description of the solution to a problem, a series of clear instructions for solving the problem, and an algorithm represents a strategy mechanism for describing the problem in a systematic way. In other words, it can obtain the required output within a limited time for a certain standard input.

CSDN says: It is a well-defined calculation process that takes one or a group of values as input and produces one or a group of values as output. In simple terms, an algorithm is a series of calculation steps used to convert input data into output results.

To put it simply: it is to turn a problem into a mathematical problem, and get one or more formulas from the problem, so as to use these formulas to get the correct output of the input data.

Dynamic Threshold Algorithm

Introduction:

According to the results of the linear CCD data collection, the sensor is very affected by the ambient light.

Therefore, we need to introduce a dynamic threshold algorithm for sampling data processing to cut the collected data and obtain the effective data.

Principle:

When there are some of the following situations in the image: shadows, uneven illumination, different contrasts in different places, sudden noise, background grayscale changes, etc., if only a fixed global threshold is used to segment the entire image, the segmentation effect will be affected because it cannot take into account the situation in various parts of the image. One solution is to use a set of thresholds related to the pixel position (that is, the threshold is a function of the coordinates) to segment each part of the image separately. This threshold related to the coordinates is also called a dynamic threshold, and this method is also called a variable threshold method, or an adaptive threshold method. The time complexity and spatial complexity of this type of algorithm are relatively large, but the anti-noise ability is strong, and it has a better effect on some images that are not easy to segment using a global threshold.

Example:

For example, an original image with uneven illumination (bright on the left and dark on the right):

If only one global threshold is selected for segmentation, the following two situations will occur, and neither can achieve satisfactory results.

(Low threshold, good effect on bright areas, but poor dark areas)

(High threshold, good effect on dark areas, but poor bright areas)

If local threshold is used, different thresholds can be selected in bright areas and dark areas respectively, so that the overall segmentation effect is more rational.

(Segmentation results of local thresholds taken in two areas)

Furthermore, if different local thresholds are used for each number, a more ideal segmentation effect can be achieved.

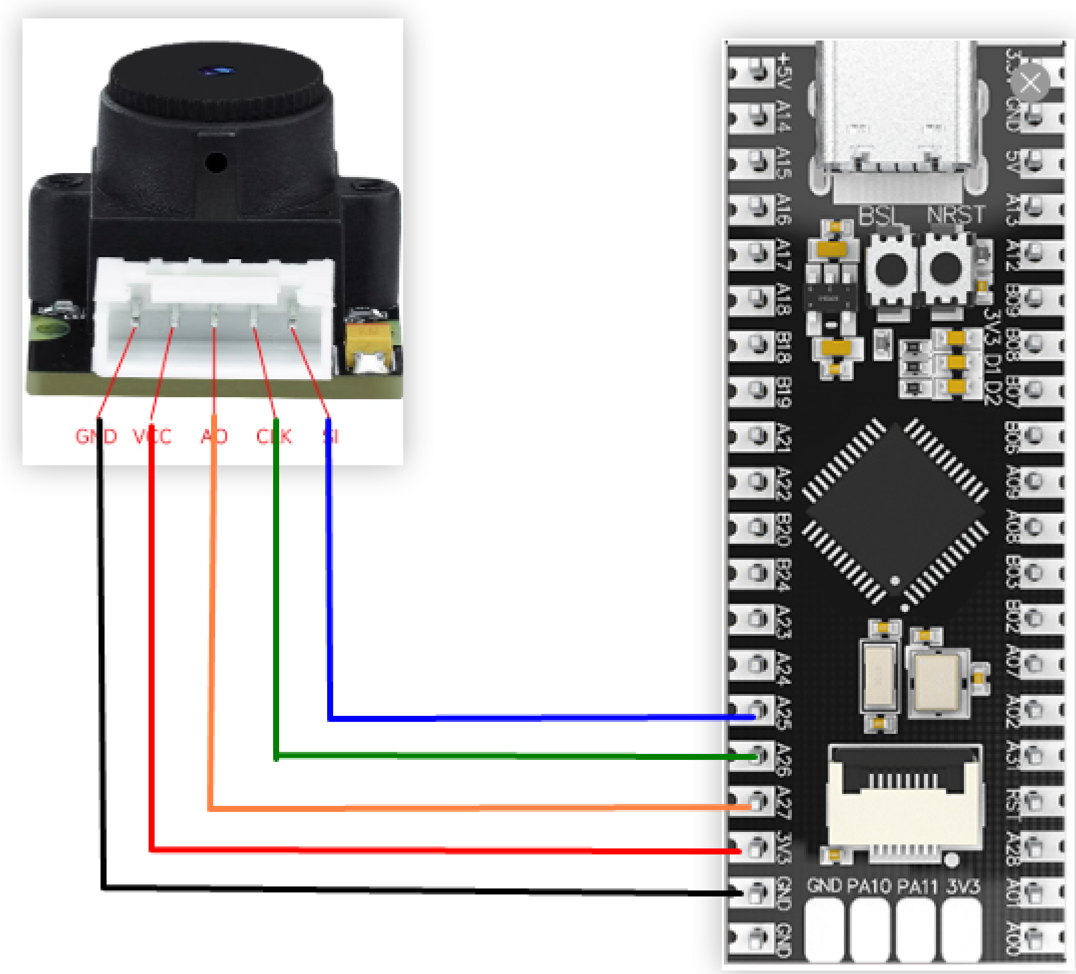
3. Hardware connection

Connection of LCD screen (need to be purchased separately) and MSPM0G3507:

MSPM0	LCD
PA31	SCLK
PA28	MOSI
PB20	RES
PB24	DC
PA23	CS
PB19	BLK

If you directly purchase our MSPM0 and the matching LCD screen, you can directly connect the LCD screen to the display interface.

Linear CCD sensor and MSPM0G3507 wiring:



4. Program Description

- ccd.c

```
uint8_t* CCD_Get_ADC_128X64(void)
{
    // 将8位AD值转化成5位AD值 Convert 8-bit AD value to 5-bit AD value
    for (int i = 0; i < 128; i++)
    {
        ADC_128X64[i] = ADV[i] >> 2;
    }
    return ADC_128X64;
}
```

A new function added here is to compress the amplitude of the returned 128 CCD data into 128*64, which just corresponds to the resolution of our LCD screen.

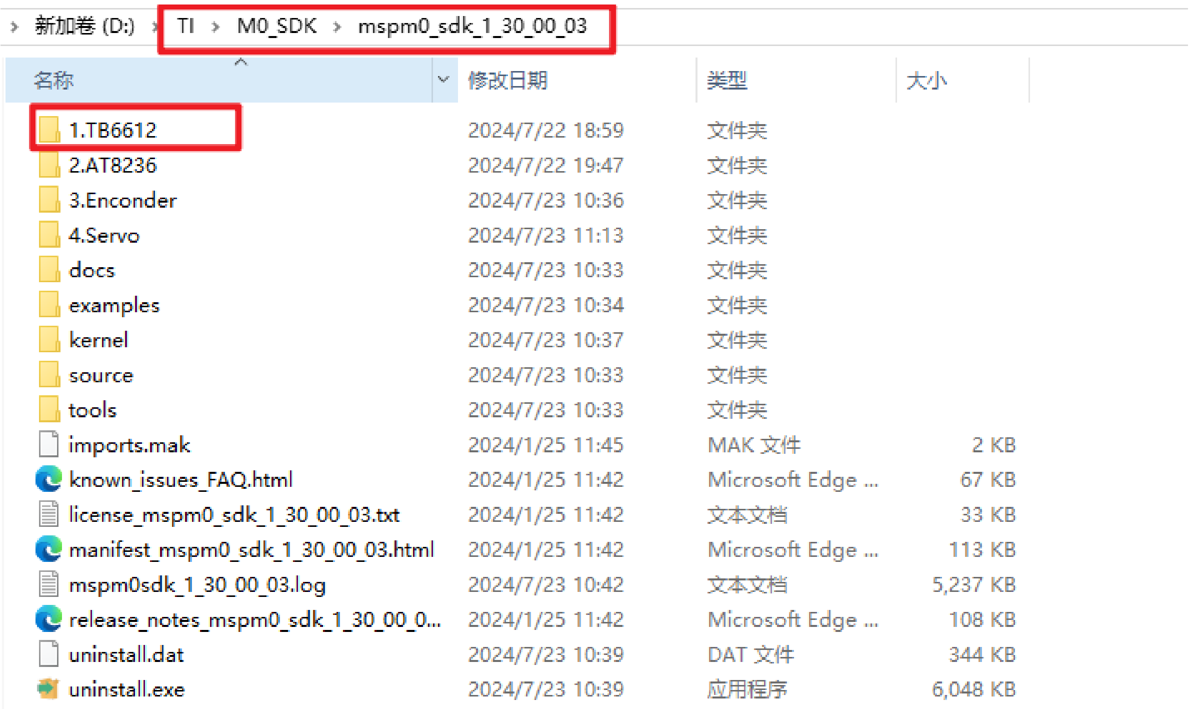
- empty.c

```
void LCD_Show_CCD_Image(uint8_t* p_img)
{
    for (int i = 0; i < 128; i++)
    {
            LCD_DrawPoint(i, p_img_last[i], BLACK);
        }


    for (int i = 15; i < 128; i++)
    {
        if (p_img[i] < 64)
        {
          LCD_DrawPoint(i, p_img[i], WHITE);
                    p_img_last[i] = p_img[i];
        }
    }

}

int main(void)
{
    USART_Init();                   //Initialization function     初始化函数

    NVIC_EnableIRQ(AO_INST_INT_IRQN);

    LCD_Init();//LCD初始化
    LCD_Fill(0,0,LCD_W,LCD_H,BLACK); //清屏

    while(1)
    {
        deal_data_ccd();
        LCD_Show_CCD_Image(CCD_Get_ADC_128X64());
    }
}
```

First, get 128 data from the CCD sensor, and then draw the waveform on the LCD (purchased separately).

Note: The project source code must be placed in the SDK path for compilation,

For example, the path: D:\TI\M0_SDK\mspm0_sdk_1_30_00_03\1.TB6612

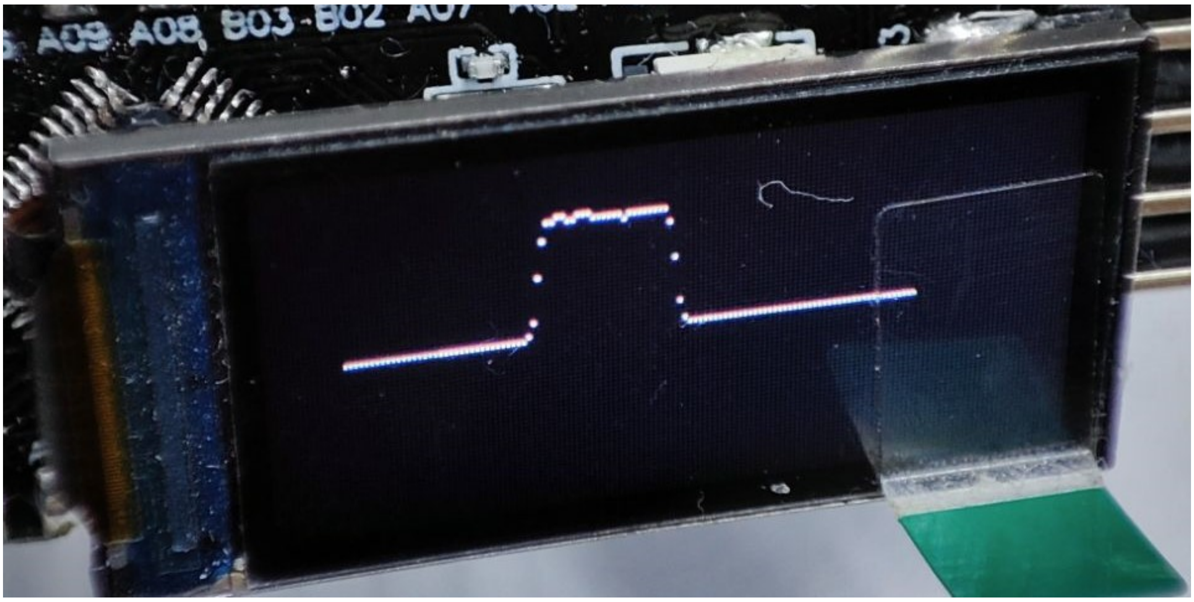


› 新加卷 (D:) › TI › M0_SDK › mspm0_sdk_1_30_00_03

名称	修改日期	类型	大小
1.TB6612	2024/7/22 18:59	文件夹	
2.AT8236	2024/7/22 19:47	文件夹	
3.Enconder	2024/7/23 10:36	文件夹	
4.Servo	2024/7/23 11:13	文件夹	
docs	2024/7/23 10:33	文件夹	
examples	2024/7/23 10:34	文件夹	
kernel	2024/7/23 10:37	文件夹	
source	2024/7/23 10:33	文件夹	
tools	2024/7/23 10:33	文件夹	
imports.mak	2024/1/25 11:45	MAK 文件	2 KB
known_issues_FAQ.html	2024/1/25 11:42	Microsoft Edge ...	67 KB
license_mspm0_sdk_1_30_00_03.txt	2024/1/25 11:42	文本文档	33 KB
manifest_mspm0_sdk_1_30_00_03.html	2024/1/25 11:42	Microsoft Edge ...	113 KB
mspm0sdk_1_30_00_03.log	2024/7/23 10:42	文本文档	5,237 KB
release_notes_mspm0_sdk_1_30_00_0...	2024/1/25 11:42	Microsoft Edge ...	108 KB
uninstall.dat	2024/7/23 10:39	DAT 文件	344 KB
uninstall.exe	2024/7/23 10:39	应用程序	6,048 KB

5. Experimental phenomenon

After connecting the wires according to the wiring diagram, burn the program to MSPM0G3507. Then you can see the following effect on the LCD screen:



By offsetting the lens, you can see that the waveform also changes, which shows that the dynamic threshold algorithm is the right choice.

It is best to experiment in an environment with only black and white lines, which will have the best effect.