Sensor algorithm processing

1. Learning objectives

Through the last experiment, we can find that the operation of the linear CCD module is seriously affected by the working environment. This experiment uses the dynamic threshold algorithm to make the CCD module unaffected by the working environment, and normally judge whether the CCD lens is facing the center line, and output the calculated value on the LCD screen (purchased separately) or the serial port assistant.

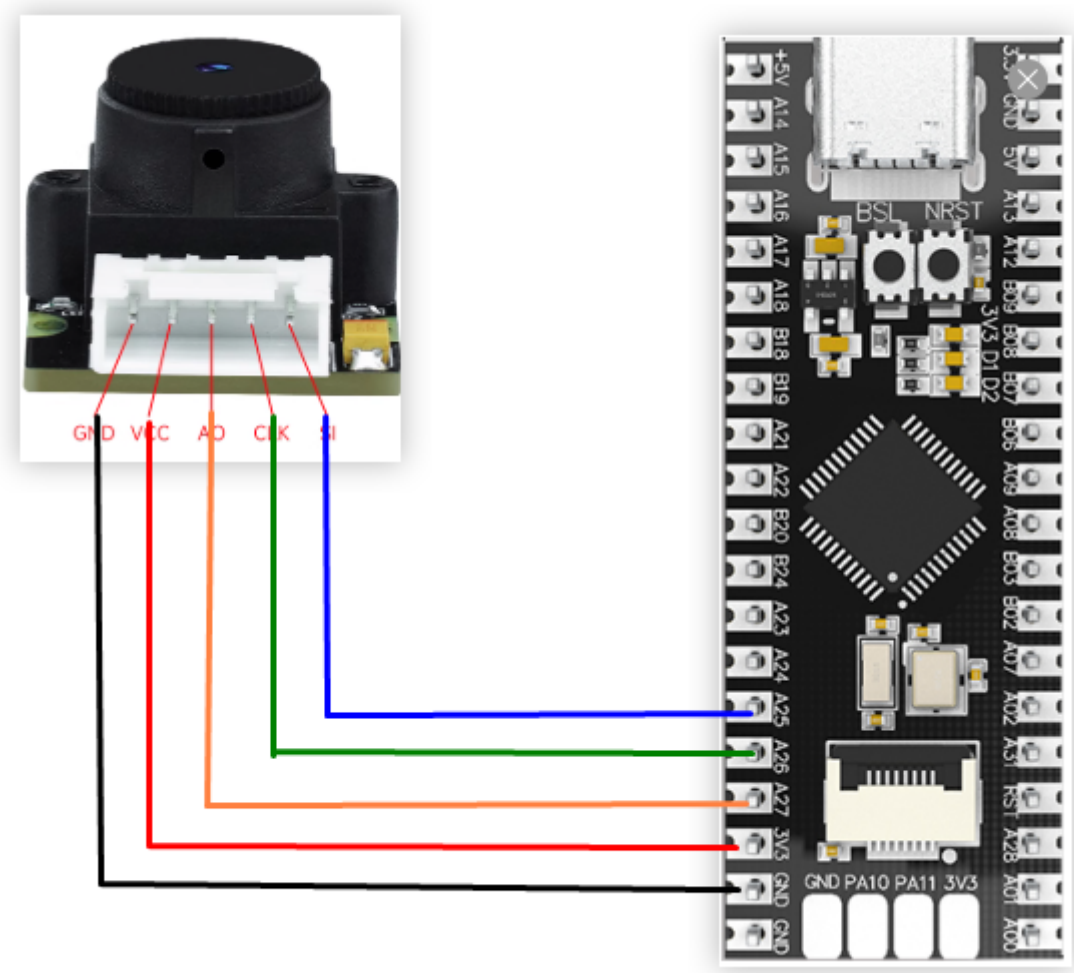
2. Hardware connection

The connection between the LCD screen (optional, needs to be purchased separately) and the 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



Note: If you don't have a TTL module, you can also use the type-c serial port directly

3. Program Description

- ccd.c

```
/*********************************************************************************
函数功能：线性CCD取中值    Function: Linear CCD median
入口参数：无            Input parameter: None
返回  值：无           Return value: None
*********************************************************************************/
void  Find_CCD_Zhongzhi(void)
{
     static uint16_t i,j,Left,Right;
     static uint16_t value1_max,value1_min;

       value1_max=ADV[0];  //动态阈值算法，读取最大和最小值     Dynamic threshold
algorithm, reads maximum and minimum values
     for(i=5;i<123;i++)   //两边各去掉5个点     Remove 5 points from each side
     {
        if(value1_max<=ADV[i])
        value1_max=ADV[i];
     }
       value1_min=ADV[0];  //最小值     Minimum
     for(i=5;i<123;i++)
     {
```

```
        if(value1_min>=ADV[i])
        value1_min=ADV[i];
     }
   CCD_Yuzhi=(value1_max+value1_min)/2;   //计算出本次中线提取的阈值    Calculate the
threshold for this midline extraction
     for(i = 5;i<118; i++)   //寻找左边跳变沿  Find the left edge
    {
        if(ADV[i]>CCD_Yuzhi&&ADV[i+1]>CCD_Yuzhi&&ADV[i+2]>CCD_Yuzhi&&ADV[i+3]
<CCD_Yuzhi&&ADV[i+4]<CCD_Yuzhi&&ADV[i+5]<CCD_Yuzhi)
        {
            Left=i;
            break;
        }
    }
     for(j = 108;j>15; j--)//寻找右边跳变沿    Find the right edge
  {
        if(ADV[j]<CCD_Yuzhi&&ADV[j+1]<CCD_Yuzhi&&ADV[j+2]
<CCD_Yuzhi&&ADV[j+3]>CCD_Yuzhi&&ADV[j+4]>CCD_Yuzhi&&ADV[j+5]>CCD_Yuzhi)
        {
          Right=j;
          break;
        }
  }
    CCD_Zhongzhi=(Right+Left)/2;//计算中线位置    Calculate the centerline position

    printf("Yuzhi: %d\r\n",CCD_Yuzhi);
    printf("Zhongzhi : %d\r\n",CCD_Zhongzhi);

    sprintf(buf_CCD,"Yuzhi:%d   ",CCD_Yuzhi);
    LCD_ShowString(0,45,(uint8_t*)buf_CCD,BLACK,WHITE,16,0);

    sprintf(buf_CCD,"Zhongzhi:%d   ",CCD_Zhongzhi);
    LCD_ShowString(0,60,(uint8_t*)buf_CCD,BLACK,WHITE,16,0);

    memset(buf_CCD,0,sizeof(buf_CCD));

}

//开始CCD采集并处理输出数据    Start CCD acquisition and process output data
void deal_data_ccd(void)
{
        RD_TSL();
        Find_CCD_Zhongzhi();
}
```

After the software starts ADC conversion, the received data is used to obtain the median and threshold through the algorithm. If you purchase a matching LCD screen, you can display the data processed by the CCD module combined with the algorithm on the display screen. If there is no LCD, you can connect the serial port to see the printed data.

CCD_Yuzhi: Extract the threshold of the center line collected this time

Use this CCD_Yuzhi threshold to determine the position of the black line in the collected data this time.

- lcd_init.h

```
#ifndef __LCD_INIT_H
#define __LCD_INIT_H

#include "ALLHeader.h"

#define USE_HORIZONTAL 2  //设置横屏或者竖屏显示 0或1为竖屏 2或3为横屏
                          //Set the screen to horizontal or vertical. 0 or 1 for
vertical screen, 2 or 3 for horizontal screen.


#if USE_HORIZONTAL==0||USE_HORIZONTAL==1
#define LCD_W 80
#define LCD_H 160

#else
#define LCD_W 160
#define LCD_H 80
#endif



//-----------------LCD端口定义----------------//
//-----------------LCD port definition---------------- //
#define LCD_SCLK_Clr() DL_GPIO_clearPins(LCD_SCLK_PORT,LCD_SCLK_PIN)//SCL=SCLK
#define LCD_SCLK_Set() DL_GPIO_setPins(LCD_SCLK_PORT,LCD_SCLK_PIN)

#define LCD_MOSI_Clr() DL_GPIO_clearPins(LCD_MOSI_PORT,LCD_MOSI_PIN)//SDA=MOSI
#define LCD_MOSI_Set() DL_GPIO_setPins(LCD_MOSI_PORT,LCD_MOSI_PIN)

#define LCD_RES_Clr()  DL_GPIO_clearPins(LCD_RES_PORT,LCD_RES_PIN)//RES
#define LCD_RES_Set()  DL_GPIO_setPins(LCD_RES_PORT,LCD_RES_PIN)

#define LCD_DC_Clr()   DL_GPIO_clearPins(LCD_DC_PORT,LCD_DC_PIN)//DC
#define LCD_DC_Set()   DL_GPIO_setPins(LCD_DC_PORT,LCD_DC_PIN)

#define LCD_CS_Clr()   DL_GPIO_clearPins(LCD_CS_PORT,LCD_CS_PIN)//CS
#define LCD_CS_Set()   DL_GPIO_clearPins(LCD_CS_PORT,LCD_CS_PIN)
//DL_GPIO_setPins(CS_PORT,CS_PIN_23_PIN)

#define LCD_BLK_Clr()  DL_GPIO_clearPins(LCD_BLK_PORT,LCD_BLK_PIN)//BLK
#define LCD_BLK_Set()  DL_GPIO_setPins(LCD_BLK_PORT,LCD_BLK_PIN)

void LCD_Writ_Bus(u8 dat);//模拟SPI时序 Simulate SPI timing
void LCD_WR_DATA8(u8 dat);//写入一个字节    Write a byte
void LCD_WR_DATA(u16 dat);//写入两个字节    Write two bytes
void LCD_WR_REG(u8 dat);//写入一个指令  Write a command
void LCD_Address_Set(u16 x1,u16 y1,u16 x2,u16 y2);//设置坐标函数    Set coordinate
function
void LCD_Init(void);//LCD初始化 LCD Initialization

#endif
```

Define the port operation of LCD, declare the basic write operation and initialization function of LCD

- lcd_init.c

```
void LCD_Init(void)
{
    LCD_RES_Clr();//复位
    delay_ms(100);
    LCD_RES_Set();
    delay_ms(100);
    LCD_BLK_Clr();//打开背光
    ....
    LCD_WR_REG(0x2B);     //Set Page Address
    LCD_WR_DATA8(0x00);
    LCD_WR_DATA8(0x01);    //1
    LCD_WR_DATA8(0x00);
    LCD_WR_DATA8(0xA0);    //160
    LCD_WR_REG(0x2C);
```

Initialize LCD

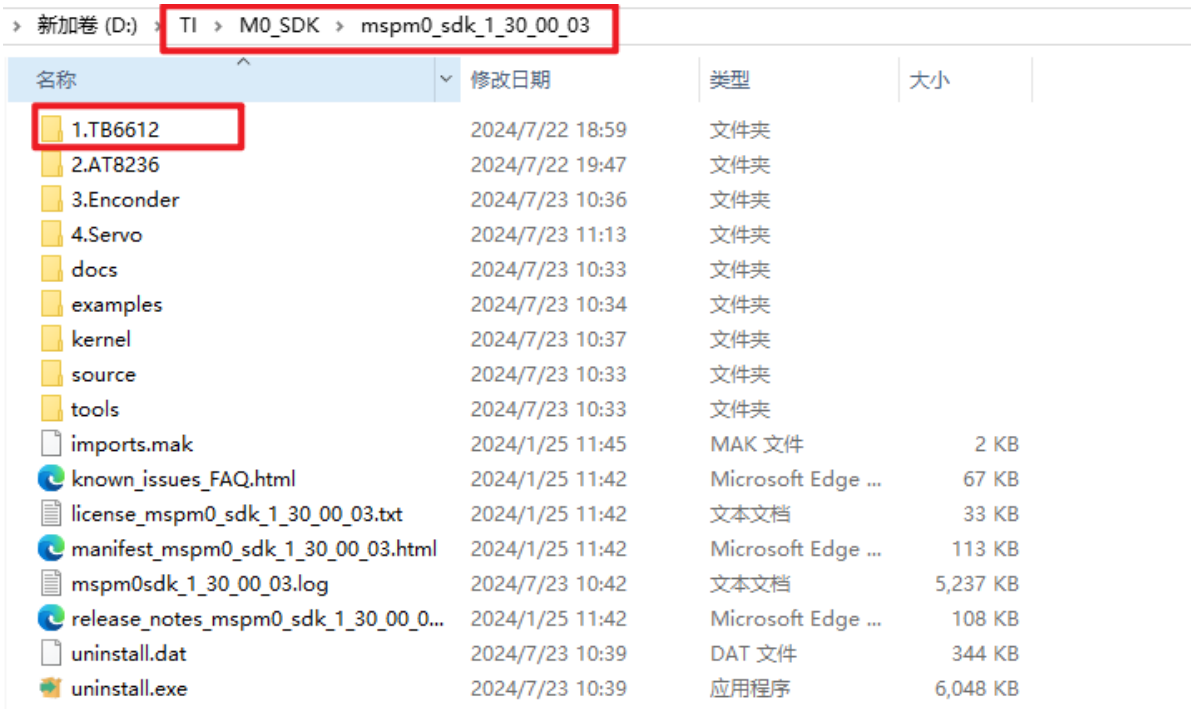
- empty.c

```
int main(void)
{
    USART_Init();   //Initialization function     初始化函数
    NVIC_EnableIRQ(AO_INST_INT_IRQN);//使能ADC中断  Enable ADC interrupt

    LCD_Init();//LCD初始化  LCD Initialization
    LCD_Fill(0,0,LCD_W,LCD_H,WHITE);

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

Call the deal_data_ccd function to obtain the CCD data and process the data through the algorithm, and finally display it on the LCD and serial port.

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

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



4. Experimental Phenomenon

After connecting the wires according to the wiring diagram, burn the program to MSPM0G3507. You can see the threshold and median data on the LCD screen, or close other programs occupying the serial port, open the serial port assistant on the computer, select the serial port number, and set the baud rate to 115200. In the serial port assistant, you can see the data printed by the CCD module.

Experimental Results

```
Yuzhi: 80
Zhongzhi : 77

[2024-12-25 11:29:53.193]# RECV ASCII>
Yuzhi: 80
Zhongzhi : 78

[2024-12-25 11:29:53.384]# RECV ASCII>
Yuzhi: 80
Zhongzhi : 77

[2024-12-25 11:29:53.575]# RECV ASCII>
Yuzhi: 80
Zhongzhi : 76

[2024-12-25 11:29:53.779]# RECV ASCII>
Yuzhi: 80
Zhongzhi : 76
```

Yuzhi: Dynamic threshold

Zhongzhi: Median

By shifting the lens, you can see that the median is changing

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