

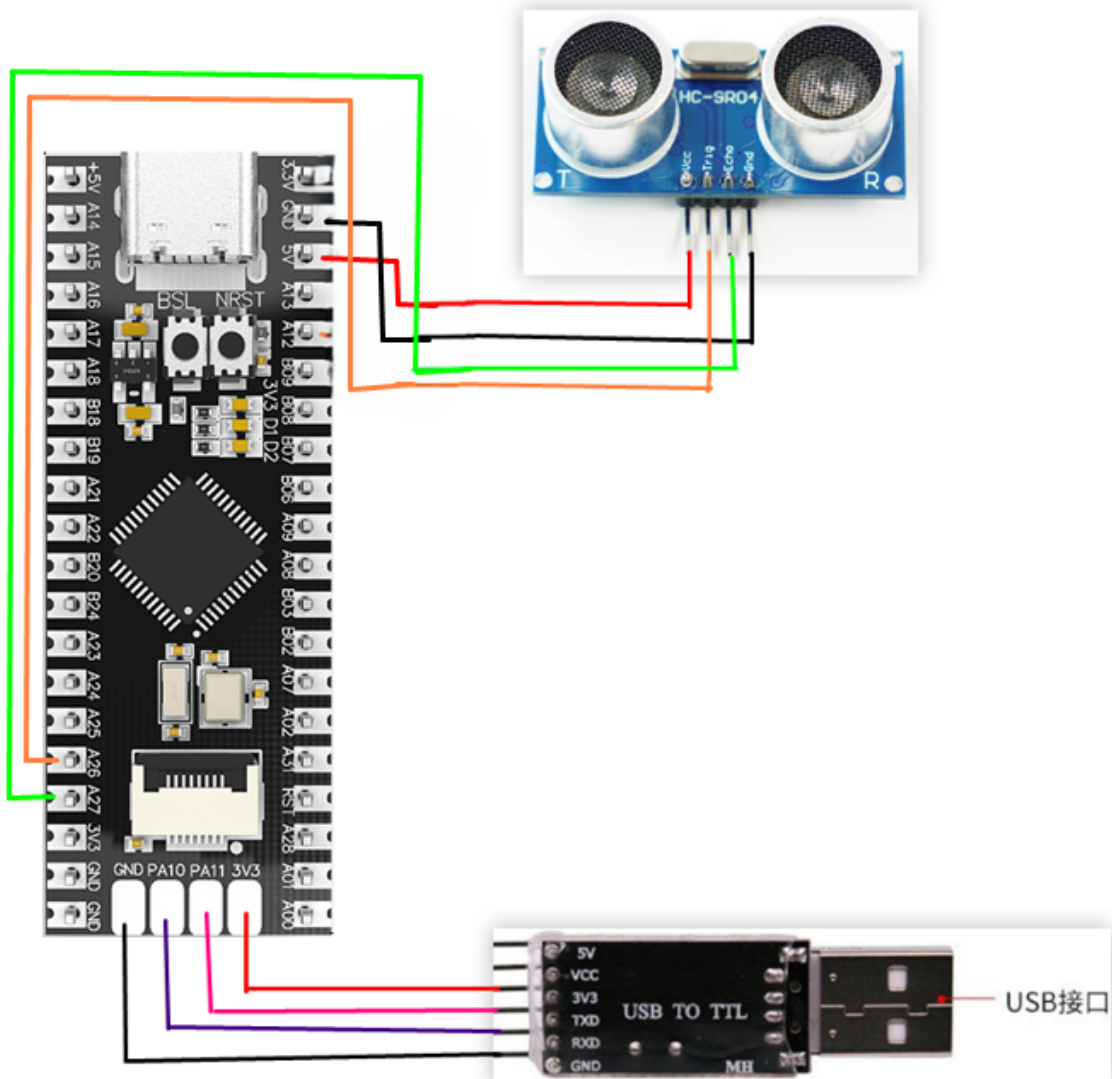
Ultrasonic distance measurement

1. Learning Objectives

The distance is measured by the ultrasonic module and printed to the computer's serial port assistant through the serial port.

2. Hardware Connection

Ultrasonic module, USB to TTL 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

- usart.c

```
void USART_Init(void)
{
    // SYSCFG初始化
    // SYSCFG initialization
```

```

SYSCFG_DL_init();
//清除串口中断标志
//Clear the serial port interrupt flag
NVIC_ClearPendingIRQ(UART_0_INST_INT_IRQN);
//使能串口中断
//Enable serial port interrupt
NVIC_EnableIRQ(UART_0_INST_INT_IRQN);
}
//串口的中断服务函数
//Serial port interrupt service function
void UART_0_INST_IRQHandler(void)
{
    uint8_t receivedData = 0;

    //如果产生了串口中断
    //If a serial port interrupt occurs
    switch( DL_UART_getPendingInterrupt(UART_0_INST) )
    {
        case DL_UART_IIDX_RX://如果是接收中断    If it is a receive interrupt

            // 接收发送过来的数据保存    Receive and save the data sent
            receivedData = DL_UART_Main_receiveData(UART_0_INST);

            // 检查缓冲区是否已满    Check if the buffer is full
            if (recv0_length < RE_0_BUFF_LEN_MAX - 1)
            {
                recv0_buff[recv0_length++] = receivedData;

                // 将保存的数据再发送出去，不想回传可以注释掉
                //Send the saved data again. If you don't want to send it back,
                you can comment it out.
                uart0_send_char(receivedData);
            }
            else
            {
                recv0_length = 0;
            }

            // 标记接收标志    Mark receiving flag
            recv0_flag = 1;

            break;

        default://其他的串口中断    Other serial port interrupts
            break;
    }
}

```

Define the initialization function of the serial port and the system, as well as the serial port receive interrupt function

- delay.c

```
#include "delay.h"
```

```

volatile unsigned int delay_times = 0;

//搭配滴答定时器实现的精确us延时
//Accurate us delay with tick timer
void delay_us(unsigned long __us)
{
    uint32_t ticks;
    uint32_t told, tnow, tcnt = 38;

    // 计算需要的时钟数 = 延迟微秒数 * 每微秒的时钟数
    // Calculate the number of clocks required = delay microseconds * number of
    clocks per microsecond
    ticks = __us * (32000000 / 1000000);

    // 获取当前的SysTick值
    // Get the current SysTick value
    told = SysTick->VAL;

    while (1)
    {
        // 重复刷新获取当前的SysTick值
        // Repeatedly refresh to get the current SysTick value
        tnow = SysTick->VAL;

        if (tnow != told)
        {
            if (tnow < told)
                tcnt += told - tnow;
            else
                tcnt += SysTick->LOAD - tnow + told;

            told = tnow;

            // 如果达到了需要的时钟数，就退出循环
            // If the required number of clocks is reached, exit the loop
            if (tcnt >= ticks)
                break;
        }
    }
}

//搭配滴答定时器实现的精确ms延时
//Accurate ms delay with tick timer
void delay_ms(unsigned long ms)
{
    delay_us( ms * 1000 );
}

```

Millisecond and microsecond delays are achieved through tick timer counting

- ultrasonic.h

```

#define SR04_TRIG(x) ( x ? DL_GPIO_setPins(TRIG_PORT,TRIG_PIN_12_PIN) :
DL_GPIO_clearPins(TRIG_PORT,TRIG_PIN_12_PIN) )
#define SR04_ECHO() ( ( ( DL_GPIO_readPins(ECHO_PORT,ECHO_PIN_13_PIN) &
ECHO_PIN_13_PIN ) > 0 ) ? 1 : 0 )

```

Macro definition controls the level of ultrasonic pin TRIG and reads the ECHO pin

- ultrasonic.c

```
//获取测量距离  Get the measured distance
volatile float t = 0;
float Hcsr04GetLength(void)
{
    /*测5次数据计算一次平均值*/
    /*Measure 5 times and calculate the average value*/
    volatile float length = 0;
    t = 0;
    volatile float sum = 0;
    volatile unsigned int i = 0;

    Close_Timer();

    while(i != 5)
    {
        SR04_TRIG(1); //TRIG引脚拉高信号，发出高电平 TRIG pin pulls up the signal
        and sends a high level

        delay_us(15); //TRIG引脚发出高电平信号10us以上 TRIG pin sends a high
        level signal for more than 10us

        SR04_TRIG(0); //TRIG引脚拉低信号，发出低电平 TRIG pin pulls down the
        signal and sends a low level

        /*Echo发出信号 等待回响信号*/
        /*输入方波后，模块会自动发射8个40KHz的声波，与此同时回波引脚（echo）端的电平会由
        0变为1;
        （此时应该启动定时器计时）；当超声波返回被模块接收到时，回波引脚端的电平会由1变为
        0;
        （此时应该停止定时器计数），定时器记下的这个时间即为超声波由发射到返回的总时长；*/

        /*Echo sends a signal and waits for the echo signal*/
        /*After inputting the square wave, the module will automatically emit
        8 40KHz sound waves. At the same time, the level of the echo pin (echo) will
        change from 0 to 1;
        (the timer should be started at this time); when the ultrasonic wave
        is returned and received by the module, the level of the echo pin will change
        from 1 to 0;
        (the timer should be stopped at this time), and the time recorded by
        the timer is the total duration of the ultrasonic wave from emission to
        return;*/

        while(SR04_ECHO() == 0); //echo等待回响 echo wait for response

        Open_Timer(); //打开定时器 Turn on the timer

        i++;

        while(SR04_ECHO() > 0);

        Close_Timer(); // 关闭定时器 Turn off the timer
```

```

        delay_ms(5);

        t = (float)Get_TIMER_Count();    // 获取时间,分辨率为1us    Acquisition
time, resolution is 1us
        length = (float)t / 58.0f;    // cm
        sum += length;
    }
    length = sum/5;//五次平均值 Five-time average
    distance = length;
    return length;
}

```

Ultrasonic acquisition distance function

- empty.c

```

int main(void)
{
    //开发板初始化    Development board initialization
    USART_Init();
    Ultrasonic_Init();
    printf("Init Ultrasonic\r\n");

    while(1)
    {
        uint32_t value = (int)Hcsr04GetLength();
        printf((const char *)"Distance = %dCM\r\n", value);
        delay_ms(5);
    }
}

```

Initialize the serial port and ultrasonic wave, and print the ultrasonic measurement data on the serial port every 5 milliseconds.

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.Encoder	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	

4. Experimental Phenomena

Burn the program to MSPM0G3507, connect the wires according to the wiring diagram. 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 9600. In the serial port assistant, you can see the printed distance data in centimeters (cm)