目前阶段完成的代码与功能如下所示：

1. 检测是否有水

#include "water.h"

#include "delay.h"

#define WA GPIO\_Pin\_11

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 配置LED用到的I/O口 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void Water\_GPIO\_Config(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure;

RCC\_APB2PeriphClockCmd( RCC\_APB2Periph\_GPIOB, ENABLE); // 使能端口时钟

GPIO\_InitStructure.GPIO\_Pin = WA;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP;

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_Init(GPIOB, &GPIO\_InitStructure); //初始化端口

//GPIO\_SetBits(GPIOB, WA); // 关闭所有LED

}

int Water\_state(GPIO\_TypeDef\* GPIOx, uint16\_t GPIO\_Pin\_x)

{

Delay\_ms(10);

if( GPIO\_ReadInputDataBit(GPIOx, GPIO\_Pin\_x) == 0)

{

return 1;

}

else

{

return 0;

}

}

if( Water\_state(GPIOB, GPIO\_Pin\_11)== 0)

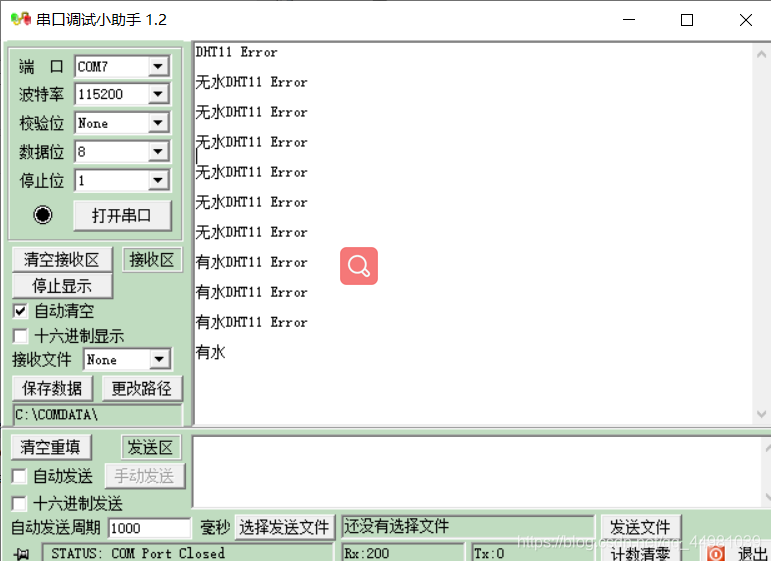
{

printf("\r\n有水");

}

else

printf("\r\n无水");



1. 测量水位

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 文件名 ：adc.c

\* 描述 ：DMA方式读取ADC值应用函数库

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "adc.h"

#define ADC1\_DR\_Address ((u32)0x4001244C)

\_\_IO u16 ADC\_ConvertedValue;

/\*配置采样通道端口 使能GPIO时钟 设置ADC采样PA0端口信号\*/

void ADC1\_GPIO\_Config(void)

{ GPIO\_InitTypeDef GPIO\_InitStructure;

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOA, ENABLE);

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_0;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_AIN; //GPIO设置为模拟输入

GPIO\_Init(GPIOA, &GPIO\_InitStructure);

}

/\*配置ADC1的工作模式为MDA模式 \*/

void ADC1\_Mode\_Config(void)

{

DMA\_InitTypeDef DMA\_InitStructure;

ADC\_InitTypeDef ADC\_InitStructure;

RCC\_AHBPeriphClockCmd(RCC\_AHBPeriph\_DMA1, ENABLE); //使能MDA1时钟

/\* DMA channel1 configuration \*/

DMA\_DeInit(DMA1\_Channel1); //指定DMA通道

DMA\_InitStructure.DMA\_PeripheralBaseAddr = ADC1\_DR\_Address;//设置DMA外设地址

DMA\_InitStructure.DMA\_MemoryBaseAddr = (u32)&ADC\_ConvertedValue; //设置DMA内存地址，ADC转换结果直接放入该地址

DMA\_InitStructure.DMA\_DIR = DMA\_DIR\_PeripheralSRC; //外设为设置为数据传输的来源

DMA\_InitStructure.DMA\_BufferSize = 1; //DMA缓冲区设置为1；

DMA\_InitStructure.DMA\_PeripheralInc = DMA\_PeripheralInc\_Disable;

DMA\_InitStructure.DMA\_MemoryInc = DMA\_MemoryInc\_Disable;

DMA\_InitStructure.DMA\_PeripheralDataSize = DMA\_PeripheralDataSize\_HalfWord;

DMA\_InitStructure.DMA\_MemoryDataSize = DMA\_MemoryDataSize\_HalfWord;

DMA\_InitStructure.DMA\_Mode = DMA\_Mode\_Circular;

DMA\_InitStructure.DMA\_Priority = DMA\_Priority\_High;

DMA\_InitStructure.DMA\_M2M = DMA\_M2M\_Disable;

DMA\_Init(DMA1\_Channel1, &DMA\_InitStructure);

/\* Enable DMA channel1 \*/

DMA\_Cmd(DMA1\_Channel1, ENABLE); //使能DMA通道

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_ADC1, ENABLE); //使能ADC1时钟

/\* ADC1 configuration \*/

ADC\_InitStructure.ADC\_Mode = ADC\_Mode\_Independent; //使用独立模式，扫描模式

ADC\_InitStructure.ADC\_ScanConvMode = ENABLE;

ADC\_InitStructure.ADC\_ContinuousConvMode = ENABLE; //无需外接触发器

ADC\_InitStructure.ADC\_ExternalTrigConv = ADC\_ExternalTrigConv\_None; //使用数据右对齐

ADC\_InitStructure.ADC\_DataAlign = ADC\_DataAlign\_Right;

ADC\_InitStructure.ADC\_NbrOfChannel = 1; // 只有1个转换通道

ADC\_Init(ADC1, &ADC\_InitStructure);

/\* ADC1 regular channel11 configuration \*/

ADC\_RegularChannelConfig(ADC1, ADC\_Channel\_0, 1, ADC\_SampleTime\_55Cycles5); //通道1采样周期55.5个时钟周期

/\* Enable ADC1 DMA \*/

ADC\_DMACmd(ADC1, ENABLE); //使能ADC的DMA

/\* Enable ADC1 \*/

ADC\_Cmd(ADC1, ENABLE); //使能ADC1

/\* Enable ADC1 reset calibaration register \*/

ADC\_ResetCalibration(ADC1);

/\* Check the end of ADC1 reset calibration register \*/

while(ADC\_GetResetCalibrationStatus(ADC1));

/\* Start ADC1 calibaration \*/

ADC\_StartCalibration(ADC1);

/\* Check the end of ADC1 calibration \*/

while(ADC\_GetCalibrationStatus(ADC1));

/\* Start ADC1 Software Conversion \*/

ADC\_SoftwareStartConvCmd(ADC1, ENABLE); //开始转换

}

/\*初始化ADC1 \*/

void ADC1\_Init(void)

{

ADC1\_GPIO\_Config();

ADC1\_Mode\_Config();

}

AD\_value = 3300000/4096\*ADC\_ConvertedValue/1000;

val=(exp(0.0056\*AD\_value))\*0.467;

printf("水位深度 = %d mm \r\n", val);

Delay\_ms(1000);



