**附录**

**一.单片机程序**

1. **主函数**

//包含显示、初始化、和传感器读数程序

int main(void)

{

uint32\_t CO2Data,TVOCData;//定义CO2浓度变量与TVOC浓度变量

uint16\_t temperature;

uint16\_t humidity;

unsigned long sgp30\_dat;

HAL\_Init();

SystemClock\_Config();

MX\_GPIO\_Init();

MX\_DMA\_Init();

MX\_USART1\_UART\_Init();

MX\_TIM6\_Init();

MX\_SPI2\_Init();

HAL\_UART\_Receive\_DMA(&huart1,aRxBuffer,1); //串口DMA初始化

OLED\_Init(); //OLED初始化

Show\_Str(0, 2,"Initializing...", 16, 0); //显示初始化中...

//DHT11初始化

while(DHT11\_Init()){

HAL\_Delay(500);

}

//初始化SGP30

SGP30\_Init();

HAL\_Delay(100);

SGP30\_Write(0x20,0x08);

sgp30\_dat = SGP30\_Read();//读取SGP30的值

CO2Data = (sgp30\_dat & 0xffff0000) >> 16;

TVOCData = sgp30\_dat & 0x0000ffff;

//SGP30模块开机需要一定时间初始化，在初始化阶段读取的CO2浓度为400ppm，TVOC为0ppd且恒定不变，因此上电后每隔500ms读取一次

//SGP30模块的值，如果CO2浓度为400ppm，TVOC为0ppd，则屏幕闪烁显示“正在检测中...”，直到SGP30模块初始化完成。

while(CO2Data == 400 && TVOCData == 0)

{

SGP30\_Write(0x20,0x08);

sgp30\_dat = SGP30\_Read();//读取SGP30的值

CO2Data = (sgp30\_dat & 0xffff0000) >> 16;//取出CO2浓度值

TVOCData = sgp30\_dat & 0x0000ffff; //取出TVOC值

HAL\_Delay(500);

}

OLED\_Clear(); //清屏

while (1)

{

DHT11\_Read\_Data(&temperature,&humidity); //读取温度湿度值

SGP30\_Write(0x20,0x08);

sgp30\_dat = SGP30\_Read();//读取SGP30的值

CO2Data = (sgp30\_dat & 0xffff0000) >> 16;//取出CO2浓度值

printf("%02d%1d",temperature>>8,temperature&0xff);

printf("%02d%1d",humidity>>8,humidity&0xff);

printf("%03d",CO2Data);

HAL\_GPIO\_TogglePin(GPIOA,GPIO\_PIN\_6);

//OLED显示

//温度显示

Show\_Str(4, 1,"temp:", 16, 0);

OLED\_ShowNum(40,1,temperature>>8,3,16);

Show\_Str(64, 1,".", 16, 0);

OLED\_ShowNum(72,1,temperature&0xff,1,16);

Show\_Str(84,1,"℃", 16, 0);

//湿度显示

Show\_Str(4, 3,"humi:", 16, 0);

OLED\_ShowNum(40,3,humidity>>8,3,16);

Show\_Str(64, 3,".", 16, 0);

OLED\_ShowNum(72,3,humidity&0xff,1,16);

Show\_Str(84, 3,"%RH", 16, 0);

//CO2浓度显示

Show\_Str(4, 5,"CO2:", 16, 0);

OLED\_ShowNum(40,5,CO2Data,3,16);

Show\_Str(70, 5,"PPM", 16, 0);

HAL\_Delay(1000); //延时1s

}

}

1. **DHT11初始化程序**

//初始化DHT11的IO口 DQ 同时检测DHT11的存在

//返回1:不存在

//返回0:存在

uint8\_t DHT11\_Init(void)

{

DHT11\_Rst();

return DHT11\_Check();

}

//复位DHT11

void DHT11\_Rst(void)

{

DHT11\_IO\_OUT(); //设置为输出

DHT11\_DQ\_OUT\_LOW; //拉低DQ

HAL\_Delay(20); //拉低至少18ms

DHT11\_DQ\_OUT\_HIGH; //DQ=1

delay\_us(30); //主机拉高20~40us

}

//等待DHT11的回应

//返回1:未检测到DHT11的存在

//返回0:存在

uint8\_t DHT11\_Check(void)

{

uint8\_t retry=0;

DHT11\_IO\_IN(); //设置为输出

while (DHT11\_DQ\_IN&&retry<100)//DHT11会拉低40~80us

{

retry++;

delay\_us(1);

};

if(retry>=100)return 1;

else retry=0;

while (!DHT11\_DQ\_IN&&retry<100)//DHT11拉低后会再次拉高40~80us

{

retry++;

delay\_us(1);

};

if(retry>=100)return 1;

return 0;

}

**3.DHT11测量程序**

//从DHT11读取一次数据

//返回值：0,正常;1,读取失败

uint8\_t DHT11\_Read\_Data(uint16\_t \*temp,uint16\_t \*humi)

{

uint8\_t buf[5];

uint8\_t i;

DHT11\_Rst();

if(DHT11\_Check()==0)

{

for(i=0;i<5;i++)//读取40位数据

{

buf[i]=DHT11\_Read\_Byte();

}

if((buf[0]+buf[1]+buf[2]+buf[3])==buf[4])

{

\*humi=(buf[0]<<8) + buf[1];

\*temp=(buf[2]<<8) + buf[3];

}

}else return 1;

return 0;

}

**4.SGP30模块初始化**

//初始化IIC接口

void SGP30\_Init(void)

{

SGP30\_Write(0x20, 0x03);

}

**5.SGP30模块测量CO2**

uint32\_t SGP30\_Read(void)

{

uint32\_t dat;

uint8\_t crc;

SGP30\_IIC\_Start();

SGP30\_IIC\_Send\_Byte(SGP30\_read); //发送器件地址+读指令

SGP30\_IIC\_Wait\_Ack();

dat = SGP30\_IIC\_Read\_Byte(1);

dat <<= 8;

dat += SGP30\_IIC\_Read\_Byte(1);

crc = SGP30\_IIC\_Read\_Byte(1); //crc数据，舍去

crc = crc; //为了不让出现编译警告

dat <<= 8;

dat += SGP30\_IIC\_Read\_Byte(1);

dat <<= 8;

dat += SGP30\_IIC\_Read\_Byte(0);

SGP30\_IIC\_Stop();

return(dat);

}

1. **Matlab GUI程序**

**1.串口接收及显示**

properties (Access = public)

COM; % 端口号

s ; %端口设置句柄

RX\_num; %接收统计

TX\_num; %发送统计

RX\_once;%一次接收的数据

RX\_Date;%接收的所以数据

is\_open;%是否打开串口标志位

end

methods (Access = public)

function EveBytesAvailableFcn(app, src, event)

global i;

n = src.BytesAvailable;%获取接收到的字符个数

if n>0%n>0才继续执行，因为0为0也会触发中断

app.RX\_num=app.RX\_num+n;

%app.Label\_RX.Text=num2str(app.RX\_num);%将数字转化为字符串输出

app.RX\_once=fread(src,n,'uchar');%读取函数，读取后矩阵为一列

app.RX\_Date =strcat(app.RX\_Date, app.RX\_once');%字符串拼接，需要转置化为一行

%app.ReceiveView.Value= app.RX\_Date;%textarea的设置格式为cell,或单行字符串

%温度显示

temp=floor(str2double(app.RX\_Date)/1000000)\*0.1;

app.Temp.Value=temp;

app.Gauge\_Temp.Value=temp;

%温度最低最高值判断

if temp>=str2double(app.TextArea\_Temp.Value)

app.Lamp\_Temp.Color="1.00,0.00,0.00";

else

app.Lamp\_Temp.Color="0.00,1.00,0.00";

end

if temp<=str2double(app.TextArea\_Temp\_2.Value)

app.Lamp\_Temp\_2.Color="1.00,0.00,0.00";

else

app.Lamp\_Temp\_2.Color="0.00,1.00,0.00";

end

%湿度显示

humi=mod((floor(str2double(app.RX\_Date)/1000)),1000)\*0.1;

app.Humi.Value=humi;

app.Gauge\_Humi.Value=humi;

%波形显示

plot(app.UIAxes,i,temp,'r.');

plot(app.UIAxes\_2,i,humi,'b.');

if i==20

i=0;

cla(app.UIAxes);

cla(app.UIAxes\_2);

else

hold(app.UIAxes,'on'); % 这样就可以hold住了

hold(app.UIAxes\_2,'on'); % 这样就可以hold住了

i=i+1;

End

%湿度最低最高值判断

if humi>=str2double(app.TextArea\_Humi.Value)

app.Lamp\_Humi.Color="1.00,0.00,0.00";

else

app.Lamp\_Humi.Color="0.00,1.00,0.00";

end

if humi<=str2double(app.TextArea\_Humi\_2.Value)

app.Lamp\_Humi\_2.Color="1.00,0.00,0.00";

else

app.Lamp\_Humi\_2.Color="0.00,1.00,0.00";

end

%CO2显示

co2=mod(str2double(app.RX\_Date),1000);

app.CO2.Value=co2;

app.Gauge\_CO2.Value=co2;

if co2>=str2double(app.TextArea\_CO2.Value)

app.Lamp\_CO2.Color="1.00,0.00,0.00";

else

app.Lamp\_CO2.Color="0.00,1.00,0.00";

end

app.RX\_Date ='';

%app.ReceiveView.Value= app.RX\_Date;

%app.TX\_num=0;

%app.RX\_num=0;

%app.Label\_TX.Text=num2str(app.TX\_num);

%app.Label\_RX.Text=num2str(app.RX\_num);

end

end

end

**2.串口接收**

function Button\_SendPushed(app, event)

%val=app.transmitView.Value;

val=strcat(strcat(app.TextArea\_Temp.Value,app.TextArea\_Humi.Value),app.TextArea\_CO2.Value);

%val=num2str(app.EditField\_Temp.Value);

if ~isempty(val{1})%textarea控件是cell格式，获取需要用{1}

app.TX\_num=app.TX\_num+length(val{1});

%app.Label\_TX.Text=num2str(app.TX\_num);

fwrite(app.s, char(val), 'uint8', 'async');%需要将val转化为char

End

**3.串口设置**

function pbOpenSerialValueChanged2(app, event)

app.COM=get(app.ppCOM,'Value');

if strcmp(get(app.pbOpenSerial,'Text'),'打开串口')

try

app.s=serial(app.COM);

app.s.BaudRate=str2num(app.Baud.Value);%设置波特率 app.s.DataBits=8;%设置数据长度

app.s.StopBits=1;%设置停止位长度

app.s.InputBufferSize=1024000;%设置输入缓冲区大小为1M

app.s.BytesAvailableFcnMode='byte'; %串口事件回调设置

app.s.BytesAvailableFcnCount=1; %输入缓冲区存在10个字节触发回调函数

app.s.BytesAvailableFcn={@app.EveBytesAvailableFcn};%回调函数的指定

fopen(app.s);%打开串口

app.is\_open=1;

app.pbOpenSerial.Text='关闭串口';

app.Lamp.Color=[0 1 0];

catch err

msgbox('打开失败');

end

else

try

fclose(app.s);

app.pbOpenSerial.Text='打开串口';

app.Lamp.Color=[0.15 0.15 0.15];

catch err

msgbox('关闭失败');

end

delete(app.s);

app.is\_open=0;

end

end











