数码管相关代码：

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

#define CLEAR 0xFFFF

#define HC595\_RCC RCC\_APB2Periph\_GPIOA

#define HC595\_PORT GPIOA

#define SLCK\_PIN GPIO\_Pin\_5

#define SCLK\_PIN GPIO\_Pin\_6

#define DATA\_PIN GPIO\_Pin\_7

//共阳端码表

static unsigned char segTable[] ={

/\*#define DB0\*/ 0xC0,

/\*#define DB1\*/ 0xF9,

/\*#define DB2\*/ 0xA4,

/\*#define DB3\*/ 0xB0,

/\*#define DB4\*/ 0x99,

/\*#define DB5\*/ 0X92,

/\*#define DB6\*/ 0x82,

/\*#define DB7\*/ 0xF8,

/\*#define DB8\*/ 0x80,

/\*#define DB9\*/ 0x90,

/\*#define DBA\*/ 0X88,

/\*#define DBB\*/ 0x83,

/\*#define DBC\*/ 0xC6,

/\*#define DBD\*/ 0xA1,

/\*#define DBE\*/ 0x86,

/\*#define DBF\*/ 0X8E

};

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

\* @brief

\* @param None

\* @retval None

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

void send2byte(unsigned short a)

{

unsigned char i,j,temp,byte;

for(j=0; j<2; j++)

{

byte = (a >>(j\*8)) & 0xff;

for(i=0;i<8;i++)

{

GPIO\_ResetBits(HC595\_PORT,SCLK\_PIN);

temp = byte & 0x80;

if (temp == 0)

{

GPIO\_ResetBits(HC595\_PORT,DATA\_PIN);

}

else

{

GPIO\_SetBits(HC595\_PORT,DATA\_PIN);

}

byte = byte << 1;

GPIO\_SetBits(HC595\_PORT,SCLK\_PIN);

}

}

}

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

\* @brief

\* @param None

\* @retval None

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

void out595(void)

{

GPIO\_ResetBits(HC595\_PORT,SLCK\_PIN);

delay\_ms(10);

GPIO\_SetBits(HC595\_PORT,SLCK\_PIN);

}

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

\* @brief

\* @param None

\* @retval None

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

void clear(void)

{

send2byte(CLEAR);

out595();

}

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

\* @brief 传感器初始化

\* @param None

\* @retval None

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

void Sensor\_Init(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure;

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

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_5|GPIO\_Pin\_6|GPIO\_Pin\_7;

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

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP; //推挽输出

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

GPIO\_SetBits(HC595\_PORT,SCLK\_PIN);

GPIO\_SetBits(HC595\_PORT,DATA\_PIN);

clear();

}

else if(strstr ( strEsp8266\_Fram\_Record.Data\_RX\_BUF, "+IPD" ) )

{

// 串口显示接收的消息

USART2\_putString((uint8\_t\*)strEsp8266\_Fram\_Record.Data\_RX\_BUF,

strEsp8266\_Fram\_Record.InfBit.FramLength);

USART2\_putString((uint8\_t\*)"\r\n", 2);

strcpy(command,strEsp8266\_Fram\_Record.Data\_RX\_BUF);

//USART2\_putString((uint8\_t\*), 1);

sprintf(str,"test %c %c %c\r\n",command[9],command[11],command[12]);

USART2\_putString((uint8\_t\*)str, strlen(str));

if((command[9]-'0')==1)//1.18

{

send2byte(segTable[(command[12]-'0')&0xf]<<8 | segTable[(command[11]-'0')&0xf]);

out595();

}

风扇：

OLED\_ShowString(0,4,"A2",16);

// 判断是否风扇控制

if((data[2] == 0x02) && (data[4] == 0x0f) && (data[7] == 0xa1))

{

buzzer\_status = data[10];

USART2\_putString("Perform Control",6);

if(data[10] == 0x01)

{

OLED\_ShowString(0,4,"openGpio",16);

GPIO\_SetBits(GPIOA, GPIO\_Pin\_6 );//开启风扇

}

else

{

OLED\_ShowString(0,4,"closeGpio",16);

GPIO\_ResetBits(GPIOA, GPIO\_Pin\_6 );//关闭风扇

}

}