#include "delay.h"

#include "sys.h"

#include "oled.h"

#include "pwm.h"

#include "delay.h"

#include "adc.h"

#include "move.h"

#include "scan.h"

#include "buzzer.h"

//#include "usart.h"

//#include "MPU6050.h"

//#include "usmart.h"

//#include "inv\_mpu.h"

#include "dma.h"

#include "timer.h"

//#include "inv\_mpu\_dmp\_motion\_driver.h"

#define TEST 0 //1为测试，0为不测试

void ltsb(void);

void delay(int t);

void TIM3\_IRQHandler(void); //TIM3中断

void DMA1\_Channel1\_IRQHandler(void);

void bianyuanshibiex(void);

void check(void);

void show(void);

void set\_num(void);

char huodejianzhi();

u8 set\_led(u8 a);

extern u32 GROUP;

extern unsigned char zheng\_or\_fan;

extern int time;

u32 temp\_z1=0,temp\_z2=0,temp\_z3=0,temp\_z4=0,temp\_z5=0,temp\_z6=0,temp\_y1=0,temp\_y2=0,temp\_y3=0,temp\_y4=0,temp\_y5=0,temp\_y6=0,temp\_Qz=0,temp\_Hz=0;

u32 temp0o=0,temp1o=0,temp2o=0,temp3o=0,temp4o=0,temp5o=0,temp6o=0,temp7o=0,temp8o=0,temp9o=0,temp\_left\_aheado=0,temp\_left\_backo=0;

u8 hui1,hui2,hui3,hui4,rlfalg=1,hui5,hui6;

u8 mm1,mm2,mm3,mm4,mm5,mm6,oldmm3,oldmm4,mm7,mm8,key=0,key\_flag=0;

u8 ltsbflag=0,allaheadflag=0,backflag=0,ltsbback=0,ltsbjiao=0,rightflag=0,leftflag=0,backflag1=0,backflag2=0,leitaichu=0,ltxback=0,ltxahead=0,ltsbin=0,enemyflag=0,uptai=0;

u32 xr=0,ra=0,la=0,yj=0,zj=0, t=0,ot=0,ot1=0,j=0,f=0,x=0,rf=0,lf=0,tj=0;

int x1,x2,x3,x4,x5,x6,x7,x0,sum[6]={0},mpu[6]={0},sub[6]={0},i,xj=0;

u16 value[14],y=0;

int kp=5,kd=1,lasttime=0,apwm=0,expectspeed=0,error=0,key\_num=0;

u32 check45\_PWM=900,check90\_PWM=900,check45\_TIME=100,check90\_TIME=200,checkback\_PWM=700,checkback\_TIME=200; //check参数设置

int main(void)

{

delay\_init(); //延时函数初始化

delay\_ms(1000);

delay\_ms(1000);

NVIC\_Configuration(); //设置NVIC中断分组2:2位抢占优先级，2位响应优先级 LED\_Init(); //LED端口初始化

BUZZER\_GPIO\_Config(); //

OLED\_Init(); //初始化OLED

MYDMA\_Config(DMA1\_Channel1,(u32)&ADC1->DR,(u32)&value,14);//初始化DMA

Adc\_Init(); //初始化ADC

TIM1\_PWM\_Init(1799,0); //电机PWM频率=72000/(1799+1)=40Khz

TIM4\_PWM\_Init(1799,0); //上台电机

TIM2\_Int\_Init(2499,143); //定时器

mm\_scan\_GPIO\_Config(); //红外对管管脚初始化（普通IO输入）

TIM3\_PWM\_Init(359\*4,0); //正交解码

while(key\_flag==0) //等待手势启动

{

set\_num();

logo(); //显示LOGO

}

OLED\_Clear() ; //OLED 清零

/\* delay\_ms(500);

ahead(1300);

aheadST(1500);

delay\_ms(220);

apwm=0;

TIM\_SetCompare3(TIM4,0);//B8 you

TIM\_SetCompare4(TIM4,0);//B9

delay\_ms(700);\*/

while(1)

{

#if TEST==1

{

TIM\_SetCompare1(TIM1,0);//a8 x/1800

TIM\_SetCompare2(TIM1,xj);//a9

TIM\_SetCompare3(TIM1,0);//a10

TIM\_SetCompare4(TIM1,xj);//a11

TIM\_SetCompare3(TIM4,0);//a10

TIM\_SetCompare4(TIM4,xj);//a11

set\_num();

show();

}

#else

{

show(); //显示参数

// ltsb(); //擂台下识别

bianyuanshibiex(); //擂台上边缘识别

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

stop();

expectspeed=0;

error=expectspeed-time;

{

apwm+=kp\*error+kd\*(error-lasttime);

if(apwm>500) {apwm=500;}

if(apwm<-500) {apwm=-500;}

if(apwm<0) {backST(-apwm);x7=2;}

else {aheadST(apwm);;x7=1;}

}

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

lasttime=error;

if(xj/160==1) {xj=0;}

set\_led(xj/20+1);

xj++;

leftflag=0;rightflag=0;backflag2=0;allaheadflag=0;ltxback=0; //flag清零

}

#endif

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*delay 改进\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

void delay(int t)

{

int i;

for(i=t/1000 ; i>0 ;i-- )

{

delay\_ms(1000);

}

delay\_ms(t%1000);

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*边缘改进\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

void bianyuanshibiex(void)

{

//doordown();

if(ltsbflag==0)

//if(hui3==1&&hui4==1&&hui1==1&&hui2==1&&mm1==1&&mm2==1) stop();

{ doordown();

/\*if(((left1>1300&&(temp\_back>1300||mm7==0||mm8==0))||(left1>1300&&(temp\_back>1300||mm7==0||mm8==0))||(left1>1300&&(temp\_right>1300||temp\_left>1300))||((temp\_back>1300||mm7==0||mm8==0)&&(temp\_right>1300||temp\_left>1300))||(temp\_ahead\_right>1300&&temp\_right>1300&&temp\_back\_right>1300)||(temp\_left>1300&&temp\_ahead\_left>1300&&temp\_back\_left>1300)))

{

//ltsbflag=1;

// ltsbin=1;

}

//else if(uptai==1){left(1000);delay\_ms(200);uptai=0;}

else\*/ if( temp\_z1>1300 ||temp\_y1>1300||temp\_Qz>1300 ) // two sensor front find enemy 2

{ //ot=5;

// if((hui1==1||hui2==1)&&GROUP==1) {back(500,500);backflag=1;}

if(mm1==1||mm3==1) {back(500,500);backflag=1;}

else

{

if(backflag==1)

{

back(500,500);

delay\_ms(100);

if(yj==1)

{

right(500);

}

else

{

left(500);

}

delay\_ms(150);

backflag=0;

}

else allahead(500);

allaheadflag=1;

}

backflag1=0;

}

else if((mm1==1||mm3==1))

{

back(500,500);backflag1=1;xj=0;backflag=0;

}

else if( temp\_z2>1300||( temp\_z3>1300)||temp\_z4>1300||temp\_z5>1300||temp\_z6>1300) //one sensor right and right back find enemy and right back 2 temp\_ahead\_right>1500 || temp\_right>1500 ||

{

left(500);

}

else if( temp\_y2>1300||( temp\_y3>1300)||temp\_y4>1300||temp\_y5>1300||temp\_y6>1300) //one sensor left and left back find enemy 2 temp\_left>1500 ||temp\_back>1500 ||

{ right(500);

}

else

{

backflag=0;

if(backflag1==1)

{

//back(500,500);delay\_ms(100);

if(yj==1)

{

right(500);

}

else

{

left(500);

}

delay\_ms(200);

backflag1=0;

}

// else if((hui1==0&&hui2==0&&hui3==0&&hui4==0)&&GROUP==1) ahead(500);//满电550

else ahead(500);

}

if(allaheadflag==1) zj++;

else zj=0;

if(ot>0) enemyflag=1;

else enemyflag=0;

if(ot<=0) ot=0;

/\*if(xj==1) BUZZER(1);

else BUZZER(0);\*/

duojiup();

ltxback=0;ltxahead=0;ot--;

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*擂台下识别\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

/\*void ltsb()

{

if(ltsbflag==1)

{ doorup();

{

if(x1>2)

{

while(value[0]>1100)

right(400);

ahead(700);

delay\_ms(500);

x1=0;

}

if((left1>1500&&(mm7==0||mm8==0)&&temp\_back\_on<1700&&temp\_left<1700&&temp\_right<1700)&&ltsbback==0&&ltsbjiao==0)

{

if(t>=3)

{

back(300,300);

duojidown();

delay\_ms(300);

back(600,600);

x1++;

delay\_ms(1000);

delay\_ms(10);

duojiup();

uptai=1;

}

t++;

}

else if((left1>1100&&temp\_right>1700&&temp\_right\_ahead>1700))

{

ahead(400);

t=0;ltxback=1;

}

else if((left1>1100&&temp\_left>1700&&temp\_left\_ahead>1700))

{

ahead(400);

t=0;ltxback=2;

}

else if((left1<1300&&temp\_left>1300&&temp\_right<1300&&temp\_back<1300&&temp\_back\_on<1300)||(left1>1300&&temp\_left<1300&&temp\_right<1300&&temp\_back<1300&&temp\_back\_on<1300)||(left1<1300&&temp\_left<1300&&temp\_right>1300&&temp\_back<1300&&temp\_back\_on<1300)||(left1<1300&&temp\_left<1300&&temp\_right<1300&&temp\_back>1300)||(left1<1300&&temp\_left<1300&&temp\_right<1300&&temp\_back<1300&&temp\_back\_on<1300))

{

if(ltsbjiao==0)

{

duojiup();

t=0;

BUZZER(0);

leitaichu++;

}

}

else

{

{

if(ltxback==1) right(400);

else left(400);

ltsbback=0;

}

duojiup();

t=0;

ltsbjiao=0;

}

if(t>0&&t<4)

{

if( temp\_back\_on>1700&&t<3) {ltxback=3;t=0;}

else if(t>1&&t<3)

{back(300,300);

delay\_ms(400);}

}

else t=0;

if(leitaichu>1)

{

leitaichu=0;

ltsbflag=0;

}

if(ltxback==1||ltxback==2) tj++;

else tj=0;

if(tj>100)

{

if(ltxback==2)

{

back(500,500);

delay\_ms(100);

while(value[0]>1100) right(400);

ahead(700);

delay\_ms(1000);

left(700);

delay\_ms(300);

tj=0;

}

else if(ltxback==1)

{

back(500,500);

delay\_ms(100);

while(value[0]>1100) left(400);

ahead(700);

delay\_ms(1000);

right(700);

delay\_ms(300);

tj=0;

}

}

if(ltxback==3)

{

if(mm7==0) while(value[0]>1100) left(400);

else while(value[0]>1100) right(400);

ahead(700);

delay\_ms(500);

}

}

}

}\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*显示与检测\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

void show()

{

temp\_z1=value[5];//

temp\_z2=value[4];//

temp\_z3=value[3];//

temp\_z4=value[2];//

temp\_z5=value[1];//

temp\_z6=value[0];//

temp\_y1=value[6];//

temp\_y2=value[7];//

temp\_y3=value[8];//

temp\_y4=value[9];//

temp\_y5=value[10];//

temp\_y6=value[11];//

temp\_Qz=value[13];//

temp\_Hz=value[12];//

/\* mm3=mm\_scan(GPIOC,GPIO\_Pin\_3); // 右下 C3 左5

mm4=mm\_scan(GPIOC,GPIO\_Pin\_1); // 左下 C1 左6

mm5=mm\_scan(GPIOD,GPIO\_Pin\_1);//默认是低 所以是下拉 下前左 D1

mm6=mm\_scan(GPIOD,GPIO\_Pin\_3); // 下前右 D3

hui3=mm\_scan(GPIOD,GPIO\_Pin\_5); // 下后左 D5

hui4=mm\_scan(GPIOD,GPIO\_Pin\_7); // 下后右 D7

mm1=mm\_scan(GPIOA,GPIO\_Pin\_13); // 默认是高 所以是上拉 右上 A13

mm2=mm\_scan(GPIOA,GPIO\_Pin\_15); // 左上 A15\*/

// hui3=mm\_scan(GPIOA,GPIO\_Pin\_13); //默认是低 所以是下拉 下后左 D6

// hui4=mm\_scan(GPIOA,GPIO\_Pin\_15); // 下后右 D7

mm1=mm\_scan(GPIOD,GPIO\_Pin\_7);//

mm2=mm\_scan(GPIOD,GPIO\_Pin\_6);

mm3=mm\_scan(GPIOD,GPIO\_Pin\_2);

mm4=mm\_scan(GPIOD,GPIO\_Pin\_1);

/\* if(temp\_z1>1300) OLED\_ShowNum(0,0,temp\_z1,4,12);

else OLED\_ShowNum(0,0,0,4,12);

if(temp\_z2>1300) OLED\_ShowNum(0,2,temp\_z2,4,12);

else OLED\_ShowNum(0,2,0,4,12);

if(temp\_z3>1300) OLED\_ShowNum(0,4,temp\_z3,4,12);

else OLED\_ShowNum(0,4,0,4,12);

if(temp\_z4>1300) OLED\_ShowNum(0,6,temp\_z4,4,12);

else OLED\_ShowNum(0,6,0,4,12);

if(temp\_z5>1300) OLED\_ShowNum(26,2,temp\_z5,4,12);

else OLED\_ShowNum(26,2,0,4,12);

if(temp\_z6>1100) OLED\_ShowNum(26,4, temp\_z6,4,12);

else OLED\_ShowNum(26,4, 0,4,12);

if(temp\_y1>1300) OLED\_ShowNum(78,0,temp\_y1,4,12);

else OLED\_ShowNum(78,0,0,4,12);

if(temp\_y2>1300) OLED\_ShowNum(78,2,temp\_y2,4,12);

else OLED\_ShowNum(78,2,0,4,12);

if(temp\_y3>1300) OLED\_ShowNum(78,4,temp\_y3,4,12);

else OLED\_ShowNum(78,4,0,4,12);

if(temp\_y4>1300) OLED\_ShowNum(78,6,temp\_y4,4,12);

else OLED\_ShowNum(78,6,0,4,12);

if(temp\_y5>1300) OLED\_ShowNum(52,2,temp\_y5,4,12);

else OLED\_ShowNum(52,2,0,4,12);

if(temp\_y6>1300) OLED\_ShowNum(52,4,temp\_y6,4,12);

else OLED\_ShowNum(52,4,0,4,12);

if(temp\_Qz>1300) OLED\_ShowNum(39,0,temp\_Qz,4,12);

else OLED\_ShowNum(39,0,0,4,12);

if(temp\_Hz>1300) OLED\_ShowNum(39,6,temp\_Hz,4,12);

else OLED\_ShowNum(39,6,0,4,12); \*/

if(apwm<0) OLED\_ShowNum(0,0,2,4,12);

else OLED\_ShowNum(0,0,1,4,12);

OLED\_ShowNum(26,0,abs(apwm),4,12);

if(time<0) OLED\_ShowNum(0,2,2,4,12);

else OLED\_ShowNum(0,2,1,4,12);

OLED\_ShowNum(26,2,abs(time),4,12);

if(error<0) OLED\_ShowNum(0,4,2,4,12);

else OLED\_ShowNum(0,4,1,4,12);

OLED\_ShowNum(26,4,abs(error),4,12);

OLED\_ShowNum(52,0,x7,1,12);

// OLED\_ShowNum(104,4,mm4,1,5);

OLED\_ShowNum(0,6,zheng\_or\_fan,1,5);

}

char huodejianzhi()

{ //1 left 2 right 3 up 4 down 5 mid

u8 key\_l=0,key\_r=0,key\_up=0,key\_down=0,key\_mid=0;

key\_l=key\_get(GPIOC,GPIO\_Pin\_6); //d13

key\_r=key\_get(GPIOD,GPIO\_Pin\_13); //d12

key\_up=key\_get(GPIOD,GPIO\_Pin\_12); //c6

key\_down=key\_get(GPIOD,GPIO\_Pin\_14); //d15

key\_mid=key\_get(GPIOD,GPIO\_Pin\_15); //d14

//OLED\_Clear() ;

/\* OLED\_ShowNum(0,2,key\_l,4,12);

OLED\_ShowNum(52,2,key\_r,4,12);

OLED\_ShowNum(26,0,key\_up,4,12);

OLED\_ShowNum(26,4,key\_down,4,12);

OLED\_ShowNum(26,2,key\_mid,4,12);\*/

if(key\_l == 1) //left

{

key=1;

}

if(key\_r == 1) //right

{

{key=2;

}

}

if(key\_up == 1) //up

{

{key=3;

}

}

if(key\_down == 1) //down

{

{key=4;

}

}

if(key\_mid == 1) //mid

{

{key=5;

}

}

return key; }

void set\_num()

{

switch(huodejianzhi())//key

{

case 5: //mid

{

key=0;

delay\_ms(150);

if(key\_flag==1)

{

key\_flag=0;

}

else if(key\_flag==0)

{

key\_flag=1;

}

delay\_ms(150);

//xj-=1;

} break;

default: break;

case 4: //down

{

key=0;

delay\_ms(120);

xj-=1;

delay\_ms(120);

} break;

case 3: //up

{

key=0;

delay\_ms(120);

xj+=1;

delay\_ms(120);

}break;

case 2: //right

if(key\_num==0)

{

key=0;

delay\_ms(120);

xj+=50;

delay\_ms(120);

}

/\* else if(key\_num==1)

{

key=0;

delay\_ms(120);

expect\_Speed\_Min-=3;

delay\_ms(120);

}

else if(key\_num==2)

{

key=0;

delay\_ms(120);

slow1-=3;

delay\_ms(120);

}

else if(key\_num==3)

{

key=0;

delay\_ms(120);

eKP-=20;

delay\_ms(120);

}

else if(key\_num==4)

{

key=0;

delay\_ms(120);

KD-=10;

delay\_ms(120);

}

else if(key\_num==5)

{

key=0;

delay\_ms(120);

SpeedKP-=1;

delay\_ms(120);

}

else if(key\_num==6)

{

key=0;

delay\_ms(120);

SpeedKI-=5;

delay\_ms(120);

}

else if(key\_num==7)

{

key=0;

delay\_ms(120);

SpeedKD-=1;

delay\_ms(120);

}\*/

break;

case 1: //left

if(key\_num==0)

{

key=0;

delay\_ms(120);

xj-=50;

delay\_ms(120);

}

/\* else if(key\_num==1)

{

key=0;

delay\_ms(120);

expect\_Speed\_Min+=3;

delay\_ms(120);

}

else if(key\_num==2)

{

key=0;

delay\_ms(120);

slow1+=3;

delay\_ms(120);

}

else if(key\_num==3)

{

key=0;

delay\_ms(120);

eKP+=20;

delay\_ms(120);

}

else if(key\_num==4)

{

key=0;

delay\_ms(120);

KD+=10;

delay\_ms(120);

}

else if(key\_num==5)

{

key=0;

delay\_ms(120);

SpeedKP+=1;

delay\_ms(120);

}

else if(key\_num==6)

{

key=0;

delay\_ms(120);

SpeedKI+=5;

delay\_ms(120);

}

else if(key\_num==7)

{

key=0;

delay\_ms(120);

SpeedKD+=1;

delay\_ms(120);

}\*/

break;

}

}

u8 set\_led(u8 a)

{

switch(a)

{

case 1:

GPIO\_ResetBits(GPIOD,GPIO\_Pin\_11); //a

GPIO\_ResetBits(GPIOD,GPIO\_Pin\_10); //b

GPIO\_ResetBits(GPIOD,GPIO\_Pin\_9); //c

break;

case 2:

GPIO\_SetBits(GPIOD,GPIO\_Pin\_11); //a

GPIO\_ResetBits(GPIOD,GPIO\_Pin\_10); //b

GPIO\_ResetBits(GPIOD,GPIO\_Pin\_9); //c

break;

case 3:

GPIO\_ResetBits(GPIOD,GPIO\_Pin\_11); //a

GPIO\_SetBits(GPIOD,GPIO\_Pin\_10); //b

GPIO\_ResetBits(GPIOD,GPIO\_Pin\_9); //c

break;

case 4:

GPIO\_SetBits(GPIOD,GPIO\_Pin\_11); //a

GPIO\_SetBits(GPIOD,GPIO\_Pin\_10); //b

GPIO\_ResetBits(GPIOD,GPIO\_Pin\_9); //c

break;

case 5:

GPIO\_ResetBits(GPIOD,GPIO\_Pin\_11); //a

GPIO\_ResetBits(GPIOD,GPIO\_Pin\_10); //b

GPIO\_SetBits(GPIOD,GPIO\_Pin\_9); //c

break;

case 6:

GPIO\_SetBits(GPIOD,GPIO\_Pin\_11); //a

GPIO\_ResetBits(GPIOD,GPIO\_Pin\_10); //b

GPIO\_SetBits(GPIOD,GPIO\_Pin\_9); //c

break;

case 7:

GPIO\_ResetBits(GPIOD,GPIO\_Pin\_11); //a

GPIO\_SetBits(GPIOD,GPIO\_Pin\_10); //b

GPIO\_SetBits(GPIOD,GPIO\_Pin\_9); //c

break;

case 8:

GPIO\_SetBits(GPIOD,GPIO\_Pin\_11); //a

GPIO\_SetBits(GPIOD,GPIO\_Pin\_10); //b

GPIO\_SetBits(GPIOD,GPIO\_Pin\_9); //c

break;

}

return 0;

}