#include"DSP28x\_Project.h"//DeviceHeaderfileandExamplesIncludeFile

/\*------------函数声明---------------\*/

interruptvoidadc\_isr(void);//ADC中断子函数

voidSetup\_Adc(void); //ADC模块配置子函数

voidSetup\_ePWM1(void); //ePWM1模块配置子函数

voidSetup\_ePWM2(void); //ePWM2模块配置子函数

/\*------------宏定义-------------\*/

#defineCPU\_CLK 150e6 //系统时钟 150MHz

#definePWM\_CLK 50e3 //开关频率 20kHz

#defineTsam 1/(2\*PWM\_CLK)//采样周期 Tsam=1/40k=25us

#defineSPCPU\_CLK/(2\*PWM\_CLK)//ePWM周期寄存器的值 3750

//PI调节器参数

#defineKp\_V 3.5 //电压环比例系数

#definetau\_V 2.91667 //电压环补偿器零点

#defineKi\_V Kp\_V/tau\_V //电压环积分系数

#defineKp\_I 2.45 //电流环比例系数

#definetau\_I 24.5 //电流环补偿器零点

#defineKi\_I Kp\_I/tau\_I //电流环积分系数

//限幅值

#defineIref\_Max 6 //电压环输出限幅

#defineIref\_Min -6

#defineVc\_Max 0.95 //电流环输出限幅

#defineVc\_Min 0.05

#defineUmax 30.1 //过压保护阈值

#defineImax 0.4 //过流保护阈值

/\*---------变量定义及初始化----------------\*/

floatUo=0; //输出电压

floatIL=0; //电感电流

floatUin=0; //输入电压

int Uo\_AD=0;

int IL\_AD=0;

int Uin\_AD=0;

floatUo\_Real[800]; //存放 20ms的输出电压

int Phase\_index=0; //数组指针

floatUref=15; //输出电压给定 Uo\*=15V

floatek\_Uo=0,ek\_Uo\_pre=0;//误差电压 e(k)和 e(k-1)

floatIref=0; //电感电流给定 IL\*

floatek\_IL=0,ek\_IL\_pre=0;//误差电流 e(k)和 e(k-1)

floatVc=0; //调制波幅值 Vc(t)

intstart\_flag=0; //启动标志，为 1表示装置正常运行

intprotect\_flag=0; //保护标志，为 1表示已启动保护

/\*-------------主函数---------------\*/

voidmain(void)

{

//Step1.初始化系统控制

InitSysCtrl(); //初始化 PLL和外设时钟

//Step2.初始化 GPIO

InitEPwm1Gpio();

EALLOW;

GpioCtrlRegs.GPAMUX1.bit.GPIO4=1;

GpioCtrlRegs.GPAMUX1.bit.GPIO5=1;//将 GPIO0和 GPIO1配置为 ePWM功

能

GpioCtrlRegs.GPADIR.bit.GPIO4=1;//将GPIO60和GPIO61配置为通用输出口

GpioCtrlRegs.GPADIR.bit.GPIO5=1;

EDIS;

GpioDataRegs.GPACLEAR.bit.GPIO0=1;//将 ePWM1引脚初始化为低电平，

此时 MOS管均关断

GpioDataRegs.GPACLEAR.bit.GPIO1=1;

GpioDataRegs.GPASET.bit.GPIO4=1;//将 GPIO4和 GPIO5初始化为低电平，此时 LED2亮，LED1灭

GpioDataRegs.GPACLEAR.bit.GPIO5=1;

//Step3.清除所有中断并初始化中断向量表

DINT; //禁用 CPU总中断

InitPieCtrl();//关闭所有 PIE模块的中断，清除所有 PIE中断标志位

IER=0x0000;//禁用 CPU中断并清除中断标志

IFR=0x0000;

InitPieVectTable();//初始化中断向量表

EALLOW;

PieVectTable.ADCINT=&adc\_isr;//将程序中需要的中断映射到中断向量表

EDIS;

//Step4.初始化外设

InitAdc();//初始化 ADC模块(使能 ADC时钟，校验 ADC，并给 ADC上电)

Setup\_Adc(); //配置 ADC模块

Setup\_ePWM1(); //配置 ePWM1模块

Setup\_ePWM2(); //配置 ePWM2模块

EPwm1Regs.TBCTR=0; //计数器同时清零

EPwm2Regs.TBCTR=0;

//Step5.使能中断

IER|=M\_INT1; //使能 CPU中断：ADCINT在第 1组中断

PieCtrlRegs.PIEIER1.bit.INTx6=1;//使能 PIE中断：ADCINT是 PIE第 1组的第

6个中断

EINT; //使能总中断 INTM

ERTM; //使能总实时中断 DBGM

//Step6.循环等待中断

for(;;)

{

asm(" NOP");

}

}

/\*---------自定义子函数----------------\*/interruptvoidadc\_isr(void)//ADC中断子函数

{

//数据读取与处理

Uo\_AD=AdcRegs.ADCRESULT0>>4;//读取 AD转换结果

IL\_AD=AdcRegs.ADCRESULT1>>4;

Uin\_AD=AdcRegs.ADCRESULT2>>4;

AdcRegs.ADCTRL2.bit.RST\_SEQ1=1;

Uo=Uo\_AD\*0.00842979-0.00265;//计算电压电流实际值

IL=IL\_AD\*0.00751752-23.697444;

Uin=Uin\_AD\*0.0107976-0.089283;

//数据存储

Uo\_Real[Phase\_index]=Uo;

Phase\_index=Phase\_index+1;

if(Phase\_index>=800)Phase\_index=0;

//过压过流检测

if((Uo>Umax)||(Uo<-Umax)||(IL>Imax)||(IL<-Imax))

{

EALLOW;

EPwm1Regs.TZFRC.bit.OST=1;//封锁 ePWM1的驱动信号

EDIS;

protect\_flag=1;

GpioDataRegs.GPBSET.bit.GPIO60=1;//LED1灭

GpioDataRegs.GPBSET.bit.GPIO61=1;//LED2灭

}

//欠压检测

if((protect\_flag==0)&&(Uin>=20)&&(Uin<=40))

{

EALLOW;

EPwm1Regs.TZCLR.bit.OST=1;//使能 PWM驱动信号

EDIS;

start\_flag=1;GpioDataRegs.GPBCLEAR.bit.GPIO60=1;//LED1亮

}

else

{

EALLOW;

EPwm1Regs.TZFRC.bit.OST=1;//封锁 PWM驱动信号

EDIS;

start\_flag=0;

GpioDataRegs.GPBSET.bit.GPIO60=1;//LED1灭

Vc=0;

ek\_Uo=0,ek\_Uo\_pre=0;

Iref=0;

ek\_IL=0,ek\_IL\_pre=0;

}

//双闭环运算

if(start\_flag==1)

{

//电压外环计算

ek\_Uo=Uref-Uo; //计算误差电压

Iref+=Kp\_V\*(ek\_Uo-ek\_Uo\_pre)+Ki\_V\*Tsam\*ek\_Uo;//计算电流指令值

ek\_Uo\_pre=ek\_Uo;

if(Iref>Iref\_Max)Iref=Iref\_Max;//对电压调节器的输出进行限幅

elseif(Iref<Iref\_Min)Iref=Iref\_Min;

//电流内环计算

ek\_IL=Iref-IL;

//计算误差电流

Vc+=Kp\_I\*(ek\_IL-ek\_IL\_pre)+Ki\_I\*Tsam\*ek\_IL;//计算控制电压

ek\_IL\_pre=ek\_IL;

if(Vc>Vc\_Max)Vc=Vc\_Max;//对电流调节器的输出进行限幅

elseif(Vc<Vc\_Min)Vc=Vc\_Min;

}EPwm1Regs.CMPA.half.CMPA=(int)(SP\*Vc);//加载比较寄存器的值

//清除中断标志

AdcRegs.ADCST.bit.INT\_SEQ1\_CLR=1;//清除 ADC的中断标志位

PieCtrlRegs.PIEACK.bit.ACK1=1;//清除第 1组的中断响应标志位，使 CPU可

以响应第 1组更多中断

return;

}

voidSetup\_Adc(void)//ADC模块配置子函数

{

//设置 ADC时钟

AdcRegs.ADCTRL3.bit.ADCCLKPS=3;//FCLK=HSPCLK/(2\*ADCCLKPS)=12.5

MHz

AdcRegs.ADCTRL1.bit.CPS=0;//ADCCLK=FCLK/(CPS+1)=12.5MHz

AdcRegs.ADCTRL1.bit.ACQ\_PS=5;//采 样 窗 每 个 通 道 采 样 时 间

=(2+ACQ\_PS)\*ADCCLK

//设置转换模式

AdcRegs.ADCTRL1.bit.CONT\_RUN=0; //启动/停止模式

AdcRegs.ADCTRL1.bit.SEQ\_OVRD=0;//禁用排序器覆盖

AdcRegs.ADCTRL1.bit.SEQ\_CASC=1; //级联排序器模式

AdcRegs.ADCTRL3.bit.SMODE\_SEL=0; //顺序采样模式

//设置采样通道

AdcRegs.ADCMAXCONV.bit.MAX\_CONV1=2;//最大采样通道数为 3

AdcRegs.ADCCHSELSEQ1.bit.CONV00=0;//设置ADCINA0作为第1个采样通

道

AdcRegs.ADCCHSELSEQ1.bit.CONV01=3;//设置ADCINA3作为第2个采样通

道

AdcRegs.ADCCHSELSEQ1.bit.CONV02=1;//设置ADCINA1作为第3个采样通

道

//设置触发方式

AdcRegs.ADCTRL2.bit.EPWM\_SOCA\_SEQ1=1;//允许 ePWM的触发信号启动

SEQ1AdcRegs.ADCTRL2.bit.INT\_ENA\_SEQ1=1;//使能 SEQ1中断

}

voidSetup\_ePWM1(void)//ePWM1模块配置子函数

{

//时基模块 TB

EPwm1Regs.TBPRD=SP; //周期值 SP=CPU\_CLK/(2\*PWM\_CLK)

EPwm1Regs.TBPHS.half.TBPHS=0; //相位为 0

EPwm1Regs.TBCTL.bit.HSPCLKDIV=0;// 时 基 时 钟

TBCLK=SYSCLKOUT=1/150M

EPwm1Regs.TBCTL.bit.CLKDIV=0;

EPwm1Regs.TBCTL.bit.PHSEN=0; //禁用相位寄存器

EPwm1Regs.TBCTL.bit.CTRMODE=2; //增减计数

//计数比较模块 CC

EPwm1Regs.CMPCTL.bit.SHDWAMODE=0;//影子装载模式

EPwm1Regs.CMPCTL.bit.LOADAMODE=2;//CTR=0或 CTR=PRD时装载

EPwm1Regs.CMPA.half.CMPA=0;//设置初始占空比为 0

//动作模块 AQ

EPwm1Regs.AQCTLA.bit.CAU=1;//向上计数且 CTR=CMPA时，ePWM1A置

低

EPwm1Regs.AQCTLA.bit.CAD=2;//向下计数且 CTR=CMPA时，ePWM1A置

高

//死区产生模块 DB

EPwm1Regs.DBCTL.bit.IN\_MODE=0; //ePWM1A是双边沿输入

EPwm1Regs.DBCTL.bit.POLSEL=2;//ePWM1A不翻转，ePWM1B翻转

EPwm1Regs.DBCTL.bit.OUT\_MODE=3;//使能双边沿延时

EPwm1Regs.DBRED=30; //上升沿延时 DBRED\*TBCLK=200ns

EPwm1Regs.DBFED=30; //下降沿延时 DBRED\*TBCLK=200ns

//错误联防模块 TZ

EALLOW;

EPwm1Regs.TZCTL.bit.TZA=2;//错误事件发生时，强制 ePWM1A低状态

EPwm1Regs.TZCTL.bit.TZB=2;//错误事件发生时，强制 ePWM1B低状态EDIS;

}

voidSetup\_ePWM2(void)//ePWM2模块配置子函数

{

//时基模块 TB

EPwm2Regs.TBPRD=SP/2; //周期值设置为 ePWM1的 1/2

EPwm2Regs.TBPHS.half.TBPHS=0; //相位为 0

EPwm2Regs.TBCTL.bit.HSPCLKDIV=0;// 时 基 时 钟

TBCLK=SYSCLKOUT=1/150M

EPwm2Regs.TBCTL.bit.CLKDIV=0;

EPwm2Regs.TBCTL.bit.PHSEN=0; //禁用相位寄存器

EPwm2Regs.TBCTL.bit.CTRMODE=2; //增减计数

//事件触发模块 ET

EPwm2Regs.ETSEL.bit.SOCAEN=1;//使能 ePWM2SOCA信号产生

EPwm2Regs.ETSEL.bit.SOCASEL=1;//当 TBCTR=0时产生 ePWM2SOCA信

号

EPwm2Regs.ETPS.bit.SOCAPRD=1;//在第 1个事件时产生 ePWM2SOCA信

号