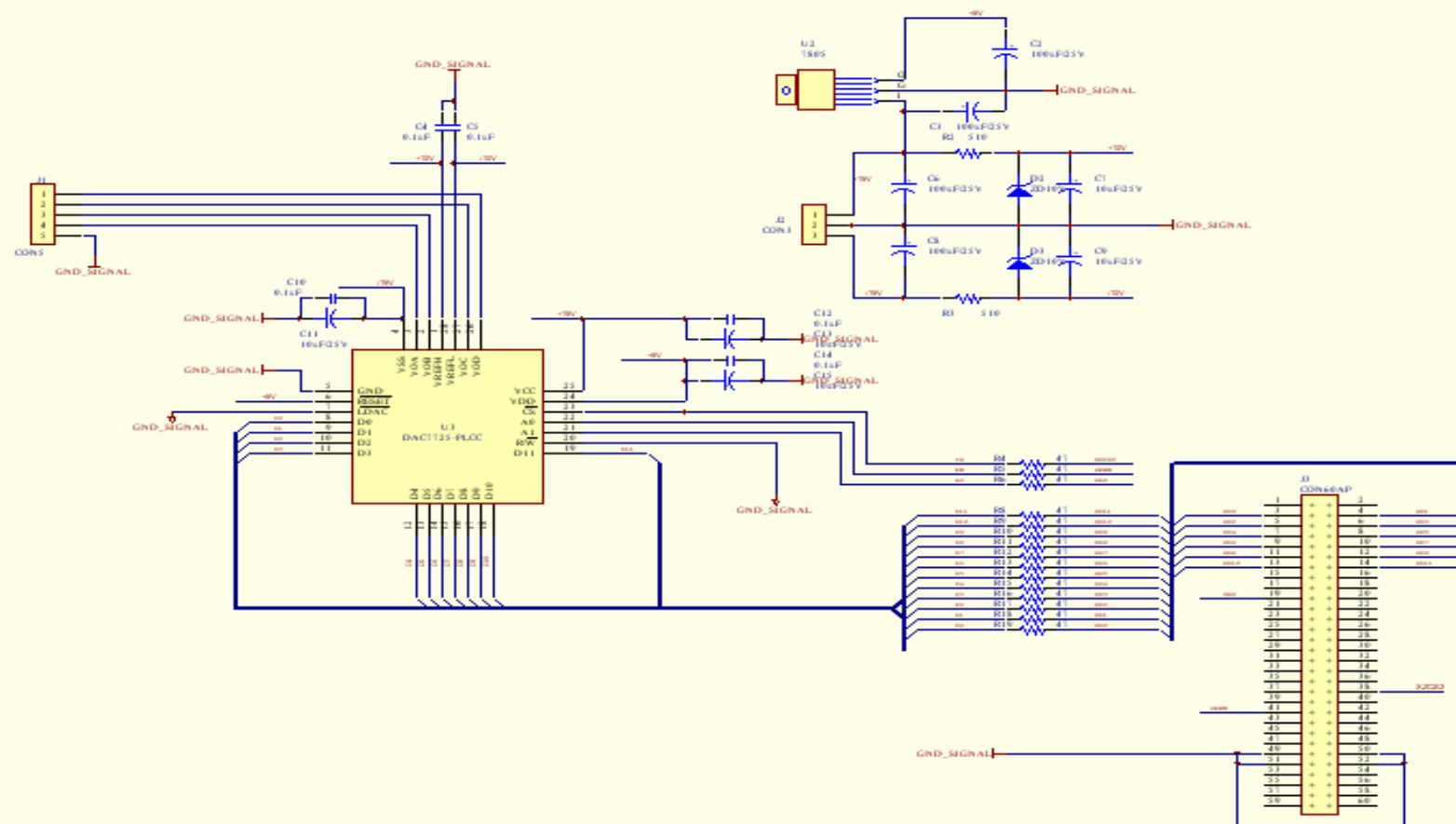


# 附錄一

## 數位類比轉換器電路





# 附錄二

## DSP 程式

```
#include "DSP28_Device.h"
#include "math.h"
#include "SinTable2000.h"
#define Kpd 256    //Q8
#define Kid 1      //Q14
#define Kpq 256    //Q8
#define Kiq 1      //Q14

interrupt void adc_isr(void);
interrupt void nmi_isr(void);

// Global variables used in this example:

far int *DA_PortA,*DA_PortB,*DA_PortC,*DA_PortD;

Uint16 Sita;
Uint16 loopindex,i,initFlag,initLoopIndex,checkOK,n=0;

Uint16 T2counterP=0,T4counterP=0,T2counter=0,T4counter=0;

float Sita_pu,Lo,iasm,ibsm,icsm;

int32 ias=0,ibs=0,ics=0,ids=0,iqs=0,iasDC=0,ibsDC=0,iasDC1=0,ibsDC1=0;
int32 SinSita,CosSita,ide,iqe;
int32 Delta_X,Delta_XP,V,V_cmd,Delta_V;
int32 Y3,U,V_cmdP,a_cmd,Fe_cmd,Fe_cmdF,iqe_cmd;
int32 ide_P,ide_I,iqe_P,iqe_I,Delta_ideP,Delta_iqeP;
int32 Delta_ide,Delta_iqe,VdeT,VqeT;
int32 VdeL,VqeL,Vde,Vqe,Vs_amp,Vds,Vqs;
int32 Vas,Vbs,Vcs,Y3P;
```

```

int32 iaDC,ibDC;
int32 X_P,X_I,V_P,V_I,Delta_XP,Delta_VP;
int32 Delta_X2,Delta_X4,iqeP,VP=0,iqeP_cmd=0;
int32 ias2=0,ibs2=0,ics2=0,iaP=0,ibP=0,icP=0,X=0,Xc=0,Xoffset=0,Vcom=0;
int32 Kpd=0,Kid=0,Kpq=0,Kiq=0,Kpx=0,Kix=0,Kpv=0,Kiv=0;

int32 ds=0,d=0,da=0,U=0,V=0,Vqes=0,Vdes=0;

main()
{
    InitSysCtrl();
    EALLOW;
    GpioMuxRegs.GPAMUX.all=0x073F;
    GpioMuxRegs.GPEMUX.bit.XNMI_XINT13_GPIOE2=1;
    EDIS;

    // Disable and clear all CPU interrupts:
    DINT;
    IER = 0x0000;
    IFR = 0x0000;

    // Initialize Pie Control Registers To Default State:
    // This function is found in the DSP28_PieCtrl.c file.
    InitPieCtrl();

    // Initialize the PIE Vector Table To a Known State:
    // This function is found in DSP28_PieVect.c.
    // This function populates the PIE vector table with pointers
    // to the shell ISR functions found in DSP28_DefaultIsr.c.
    InitPieVectTable();

    // This function is found in DSP28_InitPeripherals.c
    // InitPeripherals(); // For this example just init the ADC
    InitAdc(); // DSP28_Adc.c

    EALLOW; // This is needed to write to EALLOW protected registers
    PieVectTable.ADCINT = &adc_isr;
    PieVectTable.XNMI = &nmi_isr;

```

```

SysCtrlRegs.HISPCP.all=0x3;
EDIS;          // This is needed to disable write to EALLOW protected registers

// Include application specific functions. This is for this example:

// Enable ADCINT in PIE
PieCtrlRegs.PIEIER1.bit.INTx6 = 1;
XIntruptRegs.XNMICR.bit.POLARITY=1;
XIntruptRegs.XNMICR.bit.ENABLE=1;
XIntruptRegs.XNMICR.bit.SELECT=1;

// Enable CPU Interrupt 1
IER |= (M_INT1);          // Enable Global INT1

//variables initialize start
DA_PortA=(far int *)0x080000;
DA_PortB=(far int *)0x080001;
DA_PortC=(far int *)0x080002;
DA_PortD=(far int *)0x080003;

//variables initialize end

// Configure ADC
AdcRegs.ADCMAXCONV.all = 0x0001;          // Setup 2 conv's on SEQ1
AdcRegs.ADCCHSELSEQ1.bit.CONV00 = 0x0; // Setup ADCINA0 as 1st
SEQ1 conv.
AdcRegs.ADCCHSELSEQ1.bit.CONV01 = 0x2; // Setup ADCINA2 as 2nd
SEQ1 conv.

AdcRegs.ADCTRL2.bit.INT_ENA_SEQ1 = 1; // Enable SEQ1 interrupt
(every EOS)

// Configure EVA
// Assumes EVA Clock is already enabled in InitSysCtrl();
//EvaRegs.T1CMPR = 0x0080;          // Setup T1 compare value
EvaRegs.T1PR = 1250;              // Setup period register

```

```

EvaRegs.GPTCONA.bit.T1TOADC = 2;           // Enable EVASOC in EVA

EvaRegs.GPTCONA.bit.T1PIN=1;
EvaRegs.GPTCONA.bit.TCOMPOE=1;

EvaRegs.T1CON.all = 0x0842;           // Enable timer 1 compare (up/down mode)
EvaRegs.T1CON.bit.TPS=0;

//QEP1 setup
EvaRegs.T2CNT=0;
EvaRegs.T2CON.bit.TCLKS10=3;
EvaRegs.T2PR=0xFFFF;
EvaRegs.T2CON.bit.TMODE=3;
EvaRegs.T2CON.bit.T2SWT1=0;
EvaRegs.T2CON.bit.SET1PR=0;
EvaRegs.T2CON.bit.TENABLE=1;

EvaRegs.EXTCON.bit.QEPIE=1;
EvaRegs.EXTCON.bit.QEPIQEL=0;

//QEP2 setup
EvaRegs.T4CNT=0;
EvaRegs.T4CON.bit.TCLKS10=3;
EvaRegs.T4PR=0xFFFF;
EvaRegs.T4CON.bit.TMODE=3;
EvaRegs.T4CON.bit.T4SWT3=0;
EvaRegs.T4CON.bit.SET3PR=0;
EvaRegs.T4CON.bit.TENABLE=1;

EvaRegs.EXTCONB.bit.QEPIE=0;
EvaRegs.EXTCONB.bit.QEPIQEL=0;

//EvaRegs.EVAIMRA.bit.T1PINT=1;
//EvaRegs.EVAIFRA.bit.T1PINT=1;

///*
EvaRegs.DBTCONA.bit.EDBT1=1;
EvaRegs.DBTCONA.bit.EDBT2=1;

```

```

EvaRegs.DBTCNA.bit.EDBT3=1;
EvaRegs.DBTCNA.bit.DBTPS=4;
EvaRegs.DBTCNA.bit.DBT=10;
/**/

```

```

EvaRegs.COMCONA.all=0xA600;
EvaRegs.ACTRA.all=0x0999;

```

```

for(i=1;i<=512;i++)
{
    AdcRegs.ADCTRL2.bit.SOC_SEQ1=1;
    // AdcRegs.ADCTRL2.bit.SOC_SEQ2=1;
    while(AdcRegs.ADCST.bit.SEQ1_BSY==1);
    for(n=0;n<=4000;n++);
    while(AdcRegs.ADCST.bit.SEQ2_BSY==1);
    iasDC=iasDC+((AdcRegs.ADCRESULT0>>4)-2048);
    ibsDC=ibsDC+((AdcRegs.ADCRESULT1>>4)-2048);
    for(n=0;n<4000;n++);
}

iasDC=iasDC>>9;
ibsDC=ibsDC>>9;

```



```

AdcRegs.ADCTRL2.bit.EVA_SOC_SEQ1 = 1; // Enable EVASOC to start
SEQ1

```

```

while(1)
{
    if (checkOK==0xA8A8)
        goto next;
}
// Enable global Interrupts and higher priority real-time debug events:
next:

```

```

EINT; // Enable Global interrupt INTM
ERTM; // Enable Global realtime interrupt DBGM

```

```

while (1)

```

```

    {

    }
}

```

```

interrupt void  adc_isr(void)
{

```

```

    DINT;
    IER=0x0000;
    IFR=0x0000;

```

```

//ENTER POINT

```

```

//ADC in start

```

```

    ias2=AdcRegs.ADCRESULT0>>4;
    ibs2=AdcRegs.ADCRESULT1>>4;
    ias2=ias2-iasDC;
    ibs2=ibs2-ibsDC;
    ias2=(ias2-2048);
    ibs2=(ibs2-2048);

```



```

    ias=-ias2/2;
    ibs=-ibs2/2;
    ics=-(ias+ibs);

```

```

//ADC in end

```

```

//3-->2 start

```

```

    ids=ias<<2;
    iqs=((9459*ibs-9459*ics)>>12);
    /*DA_PortD=iqs*2000+2048;

```

```

//3-->2 end

```

```

//sin, cos start

```

```

    Sita=EvaRegs.T2CNT;

```



```

Sita_pu=(fmod(Sita,3200))/2;
if (fmod(Sita_pu,2)>0)
{

SinSita=(int)((SinTable[(int)(Sita_pu/2-0.5)]+SinTable[(int)(Sita_pu/2+0.5)])/2);

CosSita=(int)((SinTable[(int)(Sita_pu/2+400-0.5)]+SinTable[(int)(Sita_pu/2+400+0.5
)])/2);
}
else
{
    SinSita=SinTable[(Uint16)(Sita_pu)];
    CosSita=SinTable[(Uint16)(Sita_pu)+400];
}
//sin, cos end

//S--> E start
ide=((CosSita*ids+SinSita*iqs)>>14);
iqe=(-SinSita*ids+CosSita*iqs)>>14);

//S--> E end

if (loopindex>=10)
{
    loopindex=0;
//deltaX, deltaV start

if(EvbRegs.T4CNT>EvaRegs.T2CNT)
    Delta_X=(int32)(EvbRegs.T4CNT-EvaRegs.T2CNT)*10;
else
    Delta_X=-(int32)(EvaRegs.T2CNT-EvbRegs.T4CNT)*10;

T2counter=(Uint16)EvaRegs.T2CNT;
if(T2counter>T2counterP)
    V=(int32)(T2counter-T2counterP)*164;
else
    V=-(int32)(T2counterP-T2counter)*164;    //Q14

```

```

T2counterP=T2counter;
T4counterP=EvbRegs.T4CNT;

//FSSMC start
Y3=(Delta_X*170)>>1+Delta_V+(164*Y3P)>>14;
U=(Delta_X*174000)>>14+(Delta_V*4930)>>14+(Y3P*1015)>>14;
a_cmd=((V_cmd-V_cmdP)*1000)>>10;
V_cmdP=V_cmd;
Y3P=Y3;
//FSSMC end

Fe_cmd,iqe_cmd start
Fe_cmd=U +(a_cmd*9502)>>14;

iqe_cmd=(Fe_cmdF*479)>>14;

iqe_cmd=V_P+V_I; //Q14+Q14=Q14

if (iqe_cmd > 25804){ //3.15=1.575pu*Q14=25805
    iqe_cmd=25804;
}else if(iqe_cmd < -25805){
    iqe_cmd=-25805;
}

}
else
    loopindex++;

//PI start
Delta_iqe= iqe_cmd-iqe;
Delta_ide=0-ide;

iqe_P=(Delta_iqe*Kpq)>>8; //Q14
iqe_I=iqe_I+((((Delta_iqe)>>7)*((Kiq*U)>>7)));//+Delta_iqeP
//Q14          Q14          Q14

ide_P=(Delta_ide*Kpd)>>8;
ide_I=ide_I+((((Delta_ide)>>7)*((Kid*V)>>7)));//+Delta_ideP

```

//Q14

Delta\_iqueP=Delta\_ique;

Delta\_ideP=Delta\_ide;

VqeT=iqe\_P+iqe\_I; //Q14

VdeT=ide\_P+ide\_I; //Q14

//PI end

//decoupling & back-emf compensation start

Vqe=VqeT+(((V\*97)>>8)+(((V\*472)>>10)\*ide)>>18));

Vde=VdeT-(((V\*472)>>10)\*iqe)>>18);

if (Vqe > 32767){ //2pu\*Q14=32768

Vqe=32767;

U=0;

}else if(Vqe < -32768){

Vqe=-32768;

U=0;

}else{

U=1;

}



if (Vde > 32767){ //2pu\*Q14=32768

Vde=32767;

V=0;

}else if(Vde < -32768){

Vde=-32768;

V=0;

}else{

V=1;

}

//decoupling & back-emf compensation end

//limiter start

Vcom=(((Vqe>>7)\*(Vqe>>7))+((Vde>>7)\*(Vde>>7)));

```

if ( Vcom > 16384)
{
    Lo=(16384.0)/(Vcom);
    ds=(int32)(Lo*16384);

    d=((((111*(ds>>7)))>>7*((ds>>7)*(ds>>7))>>7))
      -(((ds>>7)*(ds>>7)*246)>>7)+((245*(ds>>7)))+2063;

    VqeL=((Vqe>>7)*(d>>7));
    VdeL=((Vde>>7)*(d>>7));

    //da=2000;
}
else
{
    VqeL=Vqe;
    VdeL=Vde;
    //da=-2000;
}
//limiter end

//E--> S start
Vds=((CosSita*VdeL-SinSita*VqeL)>>14);
Vqs=((SinSita*VdeL+CosSita*VqeL)>>14);

//E--> S end

//2-->3 start
Vas=Vds;
Vbs=(-8192*Vds+14188*Vqs)>>14);
Vcs=(-8192*Vds-14188*Vqs)>>14);
//2-->3 end

//SPWM start
EvaRegs.CMPR1=(int)((((Vas*625)>>14)+625);
EvaRegs.CMPR2=(int)((((Vbs*625)>>14)+625);
EvaRegs.CMPR3=(int)((((Vcs*625)>>14)+625);

```



```

//SPWM end

// Reinitialize for next ADC sequence
AdcRegs.ADCTRL2.bit.RST_SEQ1 = 1;           // Reset SEQ1
AdcRegs.ADCST    .bit.INT_SEQ1_CLR = 1;      // Clear INT SEQ1 bit
PieCtrlRegs.PIEACK.all= PIEACK_GROUP1;      // Acknowledge interrupt to
PIE
EINT;    // Enable Global interrupt INTM
ERTM;    // Enable Global realtime interrupt DBGM
return;
}

interrupt void  nmi_isr(void)
{
    EvbRegs.T4CNT=EvaRegs.T2CNT;
    Xoffset=EvaRegs.T2CNT;
    checkOK=0xA8A8;
    return;
}

```



### 附錄三

## 作者與口試委員們合影留念



(照片由左至右依序為：成功大學 陳建富教授、台灣科技大學 葉勝年教授、指導教授 潘晴財教授、作者本人。)