**实验四 AD转换**

一、实验目的

1、理解ADC12的使用方法。

2、理解串口的使用方法。

二、实验内容

将ADC12的转换结果在LCD（或LED）上显示出来。

三、实验代码

#include <msp430.h>

#include "stdint.h"

#define NumOfResult 8

#define Max\_Volt 3.3

#define Accu\_Adc 4095

uint32\_t results[NumOfResult+1];

uint16\_t average;

uint16\_t index = 0;

#define XT2\_PORT\_SEL P7SEL

#define XT2\_ENABLE (BIT2+BIT3)

// LCD segment definitions.

#define d 0x01

#define c 0x20

#define b 0x40

#define a 0x80

#define dp 0x10

#define g 0x04

#define f 0x08

#define e 0x02

uint8\_t con[10] , trans\_v[10];

const char char\_gen[] = { // As used in 430 Day Watch Demo board

a+b+c+d+e+f, // Displays "0"

b+c, // Displays "1"

a+b+d+e+g, // Displays "2"

a+b+c+d+g, // Displays "3"

b+c+f+g, // Displays "4"

a+c+d+f+g, // Displays "5"

a+c+d+e+f+g, // Displays "6"

a+b+c, // Displays "7"

a+b+c+d+e+f+g, // Displays "8"

a+b+c+d+f+g, // Displays "9"

a+b+c+e+f+g, // Displays "A"

c+d+e+f+g, // Displays "b"

a+d+e+f, // Displays "c"

b+c+d+e+g, // Displays "d"

a+d+e+f+g, // Displays "E"

a+e+f+g, // Displays "f"

a+b+c+d+f+g, // Displays "g"

c+e+f+g, // Displays "h"

b+c, // Displays "i"

b+c+d, // Displays "j"

b+c+e+f+g, // Displays "k"

d+e+f, // Displays "L"

a+b+c+e+f, // Displays "n"

a+b+c+d+e+f+g+dp // Displays "full"

};

const char char\_gen\_dp[] = { // As used in 430 Day Watch Demo board

a+b+c+d+e+f+dp, // Displays "0"

b+c+dp, // Displays "1"

a+b+d+e+g+dp, // Displays "2"

a+b+c+d+g+dp, // Displays "3"

b+c+f+g+dp, // Displays "4"

a+c+d+f+g+dp, // Displays "5"

a+c+d+e+f+g+dp, // Displays "6"

a+b+c+dp, // Displays "7"

a+b+c+d+e+f+g+dp, // Displays "8"

a+b+c+d+f+g+dp, // Displays "9"

a+b+c+e+f+g, // Displays "A"

c+d+e+f+g, // Displays "b"

a+d+e+f, // Displays "c"

b+c+d+e+g, // Displays "d"

a+d+e+f+g, // Displays "E"

a+e+f+g, // Displays "f"

a+b+c+d+f+g, // Displays "g"

c+e+f+g, // Displays "h"

b+c, // Displays "i"

b+c+d, // Displays "j"

b+c+e+f+g, // Displays "k"

d+e+f, // Displays "L"

a+b+c+e+f, // Displays "n"

g, // Displays "-"

dp, // Displays "dp"

a+b+c+d+e+f+g+dp // Displays "full"

};

/\* Private functions ---------------------------------------------------------\*/

/\*!

\*函数功能：配置LCD segment output功能

\*输入参数：无

\*返回值： 无

\*/

void Init\_lcd(void)

{

LCDBCTL0 =LCDDIV0 + LCDPRE0 + LCDMX1 + LCDSSEL + LCDMX1 + LCD4MUX ;

LCDBPCTL0 = LCDS0 + LCDS1 + LCDS2 + LCDS3 + LCDS4 + LCDS5 + LCDS6+ LCDS7 +

LCDS8

+ LCDS9 + LCDS10 + LCDS11 ;

P5SEL = 0xfc;

}

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

打开或关闭液晶

1：打开 0：关闭

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

void LcdGo(unsigned char doit)

{

if(doit==1)

{

//打开液晶显示

LCDBCTL0 |= LCDON;

}

else if(doit==0)

{

//关闭液晶显示

LCDBCTL0 &= ~LCDON;

}

}

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

显示或者消隐显示内容

doit：0：消隐 1：显示

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

void LcdBlink(unsigned char doit)

{

if(doit==0)

{

LCDBCTL0 &= ~LCDSON;

}

else if(doit==1)

{

LCDBCTL0 |= LCDSON;

}

}

void LCD\_Clear(void)//清屏

{

unsigned char index;

for (index=0; index<12; index++)

{

LCDMEM[index] = 0;

}

}

/\*port function select\*/

void Init\_TS3A5017DR(void)

{

// Configure TS3A5017DR IN1 and IN2

P1DIR |= BIT6 + BIT7; //P3.4 : IN1 ; P3.5 : IN2 set as output

P1OUT &= ~BIT7; //IN1 = 0

P1OUT |= BIT6; //IN2 = 1

}

/\*!

\*函数功能：LCD segment 背光使能

\*输入参数：无

\*返回值： 无

\*/

void Backlight\_Enable(void)

{

P8DIR |= BIT0;

P8OUT |= BIT0;

}int16\_t Trans\_val(uint16\_t number,uint16\_t system, uint8\_t \*output)

{

// 商,余数和标记

int16\_t Quotient, Balance, i = 0, j , k;

int32\_t caltmp;

uint16\_t Curr\_Volt;

caltmp = number;

caltmp = (caltmp << 5) + number; //caltmp = Hex\_Val \* 33

caltmp = (caltmp << 6) + (caltmp << 5) + (caltmp << 1); //caltmp = caltmp \* 100

Curr\_Volt = caltmp >> 12; //Curr\_Volt = caltmp / 2^n=4096

Quotient = Curr\_Volt;

do

{

Balance = Quotient % system;

output[i++] = Balance;

} while (Quotient /= system);

// 转换为char型

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

if(output[j] > 9)

output[j] = output[j] + 55;

else output[j] += 48;

// 字符串取反

for(k = 0; k < i / 2; k++)

{

char tmp;

tmp = output[k];

output[k] = output[i - k - 1];

output[i - k - 1] = tmp;

}

return i;

}

int16\_t Conversion(uint16\_t number,uint16\_t system, uint8\_t \*output)

{

// 商,余数和标记

int16\_t Quotient, Balance, i = 0,j ,k;

Quotient = number;

do

{

Balance = Quotient % system;

output[i++] = Balance;

} while (Quotient /= system);

// 转换为char型

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

if(output[j] > 9)

output[j] = output[j] + 55;

else output[j] += 48;

// 字符串取反

for(k = 0; k < i / 2; k++)

{

char tmp;

tmp = output[k];

output[k] = output[i - k - 1];

output[i - k - 1] = tmp;

}

return i;

}

void main(void)

{

WDTCTL = WDTPW+WDTHOLD; // Stop watchdog timer

Init\_TS3A5017DR(); // Configure TS3A5017DR IN1 and IN2

Init\_lcd(); // lcd初始化

LcdGo(1); // 打开液晶模块

LCD\_Clear(); //清屏

Backlight\_Enable();

P6SEL |= BIT6; // Enable A/D channel

ADC12CTL0 = ADC12ON+ADC12SHT0\_15+ADC12MSC ; // Turn on ADC12, set sampling time

// set multiple sample conversion

ADC12CTL1 = ADC12SHP + ADC12CONSEQ\_2 + ADC12SSEL\_1; // Use sampling timer, setmode ,set clock ACLK

ADC12MCTL0 |= ADC12INCH\_6; //change

ADC12IE = ADC12IE0; // Enable ADC12IFG.0

ADC12CTL0 |= ADC12ENC; // Enable conversions

ADC12CTL0 |= ADC12SC; // Start conversion

\_EINT(); // Enable all interrupt

\_BIS\_SR(LPM4\_bits + GIE); // Enter LPM4, Enable interrupts

\_\_no\_operation(); // For debugger

}

#pragma vector=ADC12\_VECTOR

\_\_interrupt void ADC12ISR (void)

{

results[index++] = ADC12MEM0; // Move results

if(index == NumOfResult)

{

uint16\_t i;

average = 0;

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

{

average += results[i];

results[i] = 0;

}

average >>= 3; //除以8

Conversion(average , 10, con);

Trans\_val(average , 10, trans\_v);

index = 0;

LCDMEM[5] = char\_gen[con[3]-48];

LCDMEM[4] = char\_gen[con[2]-48];

LCDMEM[3] = char\_gen[con[1]-48];

LCDMEM[2] = char\_gen[con[0]-48];

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

{

con[i] = 0 ;

trans\_v[i] = 0;

}

}

}

四、实验截图

ADC12的转换结果成功在LCD（或LED）上显示出来。（见附件）。

五、实验小结

本次实验，我们就对电压信号可视化做了实现验证，初始化等一系列的操作，将电阻的电压输入到单片机引脚 P6.6，在JP12接上短接帽，连接MSP430中的1和2引脚，实验板JP12左起两个引脚，通过ccs将程序读入单片机中，实现电压的动态获取，通过内部计算实现电阻的可视化。

在这次的实验中，我成功理解并掌握了ADC12的A/D转换原理并利用其来控制并转换电压。最重要的一点是，我在这次实验中培养了自己的学习能力，并巩固了课堂上学习的有关A/D转换的知识，令我受益匪浅。