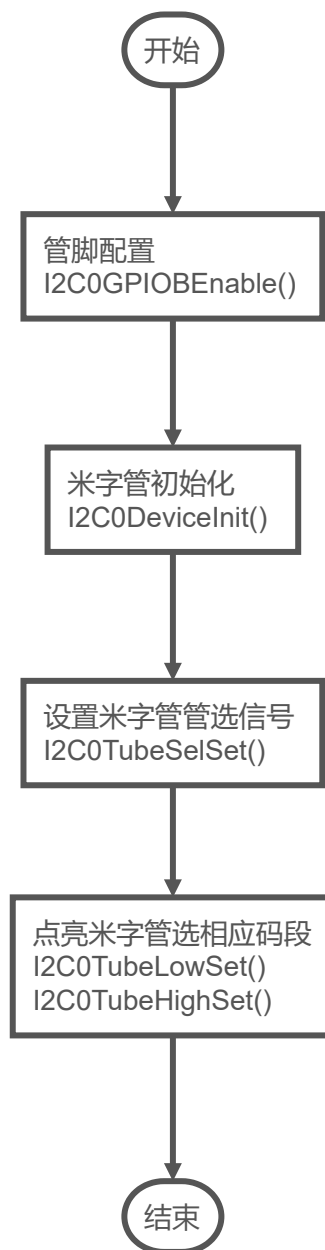


实验五

- 实验目的
 1. 了解 I2C 总线的特点和功能。
 2. 学会 C 语言模拟 I2C 时序。
 3. 学会用 I2C 总线对 PCA9557 芯片进行操作。
- 实验内容
 1. 能够点亮并且控制米字管。
 2. 能够使用ADC获得温度和电压。
- 实验流程图



- 实验代码

```
//配置I2C0模块的IO引脚，使其工作于开漏模式下  
void I2C0PinConfig(void)
```

```

{ // Enable GPIO portB containing the I2C pins (PB2&PB3)
  ui32SysClock = SysCtlClockFreqSet((SYSCTL_XTAL_25MHZ |
                                     SYSCTL_OSC_MAIN |
                                     SYSCTL_USE_PLL |
                                     SYSCTL_CFG_VCO_480),
                                     1000000); //设置系统时间为1MHZ,不能太高,会
出问题。

  SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
  GPIOPinTypeI2C(GPIO_PORTB_BASE, GPIO_PIN_2 | GPIO_PIN_3);
  GPIOPinConfigure(GPIO_PB2_I2C0SCL);
  GPIOPinConfigure(GPIO_PB3_I2C0SDA);

  //Configure the PB2 and PB3 pins for I2C operation.
  GPIOPinTypeI2CSCL(GPIO_PORTB_BASE, GPIO_PIN_2);
  GPIOPinTypeI2C(GPIO_PORTB_BASE, GPIO_PIN_3);
}

void I2C0DeviceInit(void)
{
  char dataBuf[2] = {PCA9557_REG_CONFIG, 0x00};
  I2C0Send(I2C0_ADDR_TUBE_SEL, dataBuf, 2); //U21管选
  I2C0Send(I2C0_ADDR_TUBE_SEG_LOW, dataBuf, 2); //U22
  I2C0Send(I2C0_ADDR_TUBE_SEG_HIGH, dataBuf, 2); //U23
}

//设置米字管的管选信号
void I2C0TubeSelSet(char data)
{
  char dataBuf[2] = {PCA9557_REG_OUTPUT, data};
  I2C0Send(I2C0_ADDR_TUBE_SEL, dataBuf, 2);
}

//点亮米字管的相应码段
void I2C0TubeLowSet(char data)
{
  char dataBuf[2] = {PCA9557_REG_OUTPUT, data};
  I2C0Send(I2C0_ADDR_TUBE_SEG_LOW, dataBuf, 2);
}

void I2C0TubeHighSet(char data)
{
  char dataBuf[2] = {PCA9557_REG_OUTPUT, data};
  I2C0Send(I2C0_ADDR_TUBE_SEG_HIGH, dataBuf, 2);
}

//封装好的数码管初始化函数。
void PCA9557_Init(void)
{
  I2C0PinConfig();
  I2C0MasterInit();
  I2CSlaveEnable(I2C0_MASTER_BASE);
  I2C0DeviceInit();
}

//码表
static const char tubeCodeTable[16][2]=
{ // SegmLow, SegmHigh
  { 0x10, 0x3E }, // 0
  { 0x00, 0x18 }, // 1
  { 0x70, 0x2C }, // 2
  { 0x70, 0x26 }, // 3
  { 0x60, 0x32 }, // 4
  { 0x70, 0x16 }, // 5
  { 0x70, 0x1E }, // 6
  { 0x00, 0x26 }, // 7
  { 0x70, 0x3E }, // 8
  { 0x70, 0x36 }, // 9

```

```

    { 0x60, 0x3E }, // A
    { 0x70, 0x3E }, // B
    { 0x10, 0x1C }, // C
    { 0x10, 0x3E }, // D
    { 0x70, 0x1C }, // E
    { 0x60, 0x1C }  // F
};

```

//设置数码管

```

void setnumber(int index,char value)
{
    switch(value) {
        case '0': {
            a[0]=tubeCodeTable[0][0];
            a[1]=tubeCodeTable[0][1];
            break;
        }
        case '1': {
            a[0]=tubeCodeTable[1][0];
            a[1]=tubeCodeTable[1][1];
            break;
        }
        case '2': {
            a[0]=tubeCodeTable[2][0];
            a[1]=tubeCodeTable[2][1];
            break;
        }
        case '3': {
            a[0]=tubeCodeTable[3][0];
            a[1]=tubeCodeTable[3][1];
            break;
        }
        case '4': {
            a[0]=tubeCodeTable[4][0];
            a[1]=tubeCodeTable[4][1];
            break;
        }
        case '5': {
            a[0]=tubeCodeTable[5][0];
            a[1]=tubeCodeTable[5][1];
            break;
        }
        case '6': {
            a[0]=tubeCodeTable[6][0];
            a[1]=tubeCodeTable[6][1];
            break;
        }
        case '7': {
            a[0]=tubeCodeTable[7][0];
            a[1]=tubeCodeTable[7][1];
            break;
        }
        case '8': {
            a[0]=tubeCodeTable[8][0];
            a[1]=tubeCodeTable[8][1];
            break;
        }
        case '9': {
            a[0]=tubeCodeTable[9][0];
            a[1]=tubeCodeTable[9][1];
            break;
        }
        case 'A': {
            a[0]=tubeCodeTable[10][0];

```

```

        a[1]=tubeCodeTable[10][1];
        break;
    }

    case 'B': {
        a[0]=tubeCodeTable[11][0];
        a[1]=tubeCodeTable[11][1];
        break;
    }
    case 'C': {
        a[0]=tubeCodeTable[12][0];
        a[1]=tubeCodeTable[12][1];
        break;
    }
    case 'D': {
        a[0]=tubeCodeTable[13][0];
        a[1]=tubeCodeTable[13][1];
        break;
    }
    case 'E': {
        a[0]=tubeCodeTable[14][0];
        a[1]=tubeCodeTable[14][1];
        break;
    }
    case 'F': {
        a[0]=tubeCodeTable[15][0];
        a[1]=tubeCodeTable[15][1];
        break;
    }
    case ' ': {
        a[0]=0x10;
        a[1]=0x3E;
        break;
    }
}

switch(index)
{
case(1): {
    I2C0TubeSelSet(~0x20);
    I2C0TubeSelSet(~0x20);
    break;
}
case(2): {
    I2C0TubeSelSet(~0x02);
    I2C0TubeSelSet(~0x02);
    break;
}
case(3): {
    I2C0TubeSelSet(~0x04);
    I2C0TubeSelSet(~0x04);
    break;
}
case(4): {
    I2C0TubeSelSet(~0x08);
    I2C0TubeSelSet(~0x08);
    break;
}
default:
    break;
}

```

```

    I2C0TubeLowSet(a[0]);
    I2C0TubeHighSet(a[1]);
}

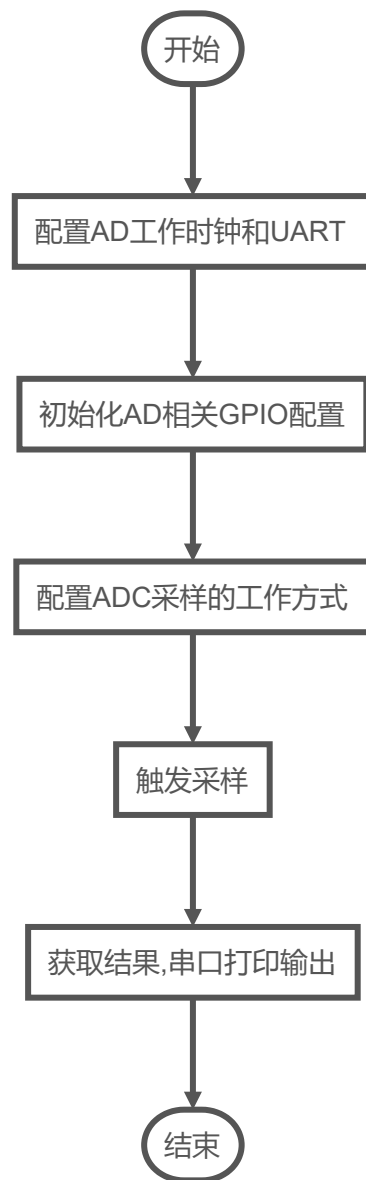
//测试函数
void PCA9557_Test()
{
    while(1)
    {
        setnumber(1,'C');
        delay();
        setnumber(2,'D');
        delay();
        setnumber(3,'E');
        delay();
        setnumber(4,'F');
        delay();
    }
}

```

- 实验现象
数码管会依次显示C,D,E,F
- 思考题
 1. 以上代码以实现.
 2. 以上代码便是用库函数的方式实现的.

实验六

- 实验目的
 1. 理解 TM4C1294KCPDT 的 ADC 模块原理。
 2. 理解 ADC 如何启动，如何判断 ADC 转换结束。
 3. 学会 ADC 模块的设计
- 实验内容
控制ADC0分别从通道0和通道1读取数据,并转化成温度和电压值并输出.
- 实验流程图



- 实验代码

```
//ADC初始化
void ADC0_Init()
{
    SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
    GPIOPinTypeADC(GPIO_PORTE_BASE, GPIO_PIN_3);
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);
    GPIOPinTypeADC(GPIO_PORTD_BASE, GPIO_PIN_7);

    ADCSequenceConfigure(ADC0_BASE, 3, ADC_TRIGGER_PROCESSOR, 0);
    ADCSequenceStepConfigure(ADC0_BASE, 3, 0, ADC_CTL_CH0 | ADC_CTL_END |
ADC_CTL_IE);
    ADCSequenceEnable(ADC0_BASE, 3);
    ADCIntClear(ADC0_BASE, 3);

    ADCSequenceConfigure(ADC0_BASE, 2, ADC_TRIGGER_PROCESSOR, 0);
    ADCSequenceStepConfigure(ADC0_BASE, 2, 0, ADC_CTL_CH4 | ADC_CTL_END |
ADC_CTL_IE);
    ADCSequenceEnable(ADC0_BASE, 2);
    ADCIntClear(ADC0_BASE, 2);
}

//ADC读取指定通道函数
```

```

uint16_t ADC0_Read(int channel)
{
    uint32_t v;
    switch(channel)
    {
        case (0):
        {
            ADCProcessorTrigger(ADC0_BASE, 3);
            while(!ADCIntStatus(ADC0_BASE, 3, false)) ;
            ADCIntClear(ADC0_BASE, 3);
            ADCSequenceDataGet(ADC0_BASE, 3, &v);
            break;
        }
        case (4):
        {
            ADCProcessorTrigger(ADC0_BASE, 2);
            while(!ADCIntStatus(ADC0_BASE, 2, false)) ;
            ADCIntClear(ADC0_BASE, 2);
            ADCSequenceDataGet(ADC0_BASE, 2, &v);
            break;
        }
    }
    return v;
}

//ADC测试
void ADC0_Test()
{
    uint16_t adc_value[10];
    float v[10];
    char s[10];
    float temperature[10];
    float temperature_avg,v_avg;
    float temp_sum;
    while(1)
    {
        for(int i=0;i<10;i++)
        {
            adc_value[i] = ADC0_Read(0);
            temperature[i] = (1.8663-((adc_value[i]/4096.0)*3.3))*1000/11.69;
            adc_value[i] = ADC0_Read(4);
            v[i] = adc_value[i]/4096.0*3.3*1000;
        }

        for(int i=0;i<10;i++)
        {
            temp_sum+=temperature[i];
        }

        temperature_avg = temp_sum/10;
        temp_sum = 0;

        for(int i=0;i<10;i++)
        {
            temp_sum+=v[i];
        }

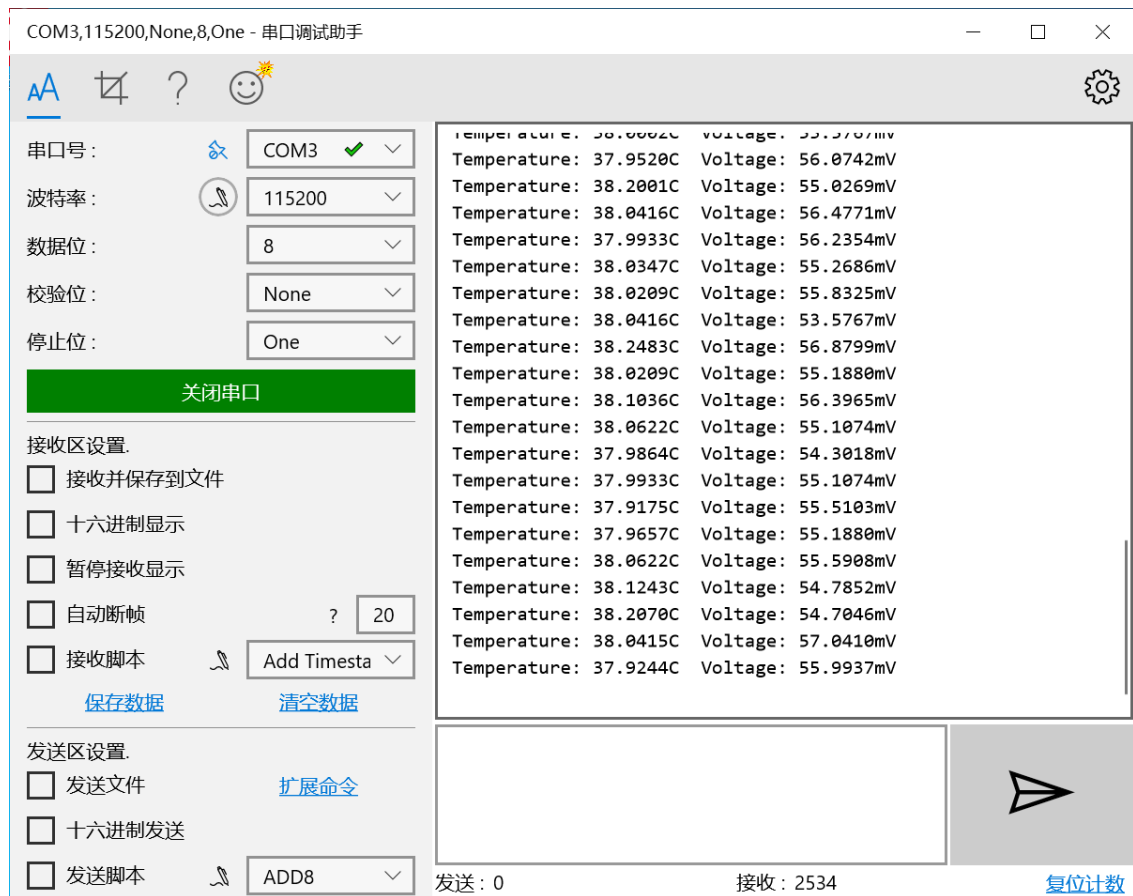
        v_avg = temp_sum/10;
        temp_sum = 0;

        sprintf(s,"Temperature: %.4lfC  Voltage:
%.4lfmV\n",temperature_avg,v_avg);
        UARTprintf(s);
    }
}

```

}

- 实验现象



- 思考题

温度的已经实现了,然后使用定时器中断定时采样温度,并用数码管显示的代码如下

```
void Timer0BIntHandler(void)
{
    //volatile uint32_t i;
    unsigned long Status;
    TimerDisable(TIMERO_BASE, TIMER_B);
    Status=TimerIntStatus(TIMERO_BASE,true);

    if(Status==TIMER_TIMB_TIMEOUT)
    {
        float v;
        char s[4];
        v = (1.8663-((ADC0_Read(0)/4096.0)*3.3))*1000/11.69*100;
        sprintf(s,"%4d",(int)v);
        int t = 2000000;

        SysCtlDelay(t);
        setnumber(1,s[0]);

        SysCtlDelay(t);

        setnumber(2,s[1]);

        SysCtlDelay(t);

        setnumber(3,s[2]);

        SysCtlDelay(t);
    }
}
```



```
        setnumber(4,s[3]);

        SysCtlDelay(t);

    }
    TimerIntClear(TIMER0_BASE, Status);
    TimerLoadSet(TIMER0_BASE, TIMER_B, g_ui32SysClock/10000 );
    TimerEnable(TIMER0_BASE, TIMER_B);
}
```