

实验四:C 语言开发/模数转换 (ADC) 实验

1. 实验目的

- 掌握 C 语言嵌入式开发的基本步骤和原理。
- 学习使用 C 语言操作外设的方法。
- 掌握模数转换器外设的操作原理和编程。

2. 实验设备

- 硬件：PC 机一台
Mini2440 ARM 实验板一套
J-link 仿真器一套
- 软件：WindowsXP 系统，Keil uVision 4.0 集成开发环境

3. 实验内容

- (1) 建立汇编-C 程序框架。
- (2) 使用 ADC 读取 Mini2440 实验板上的滑动变阻器产生的电压值。

4. 实验预习要求

- (1) 学习 ARM 子程序调用相关的指令；
- (2) 查阅 S3C2440 芯片手册，了解模数转换器系统的结构和原理。

5. 实验步骤

(一) C 语言编程实验：

(1) 实验程序分为汇编语言和 C 语言两部分。汇编语言程序是处理器上电复位后执行的第一段程序，负责最基础的硬件初始化，在本实验中主要是初始化栈指针 (SP) 寄存器，为调用 C 语言函数做准备，然后跳转到 C 语言 main 函数执行；从 main 函数开始，程序的编写结构和执行原理基本上与 PC 机上典型的 C 语言相同。

(2) 启动 Keil uVision，新建一个工程 ex04-1。不需要系统提供的 Startup 文件。建立汇编源文件 ex04-1.s，编写实验程序，然后添加到工程中。设置工程选项，存储器映射。设置工程调试选项。建立仿真初始化文件 RAM.ini。

具体步骤参考实验二。

(3) 建立 C 语言源文件 main.c，编写实验程序，然后添加到工程中。。

(4) 编译程序，使用仿真器在目标板上调试运行程序，使用单步、设置断点，观察程序执行的流程，观察函数调用过程中栈指针寄存器 (R13, SP) 的值的變化。

(5) 编程并观察 C 语言语句编译后对应的 ARM 汇编程序。

(二) 模数转换器 (ADC) 实验：

(1) 本实验使用 S3C2440 片内模数转换器，见图 4-1。从结构图和芯片手册可以知道，该 ADC 模块总共有 8 个通道可以进行模拟信号的输入，分别是 AINO、

AIN1、AIN2、AIN3、YM、YP、XM、XP。模拟信号从任一通道输入，然后设定寄存器中预分频器的值来确定 AD 转换器频率，最后 ADC 将模拟信号转换为数字信号保存到 ADC 数据寄存器 0 中(ADCDAT0)，然后 ADCDAT0 中的数据可以通过中断或查询的方式来访问。对于 ADC 的各寄存器的操作和注意事项请参阅数据芯片手册。

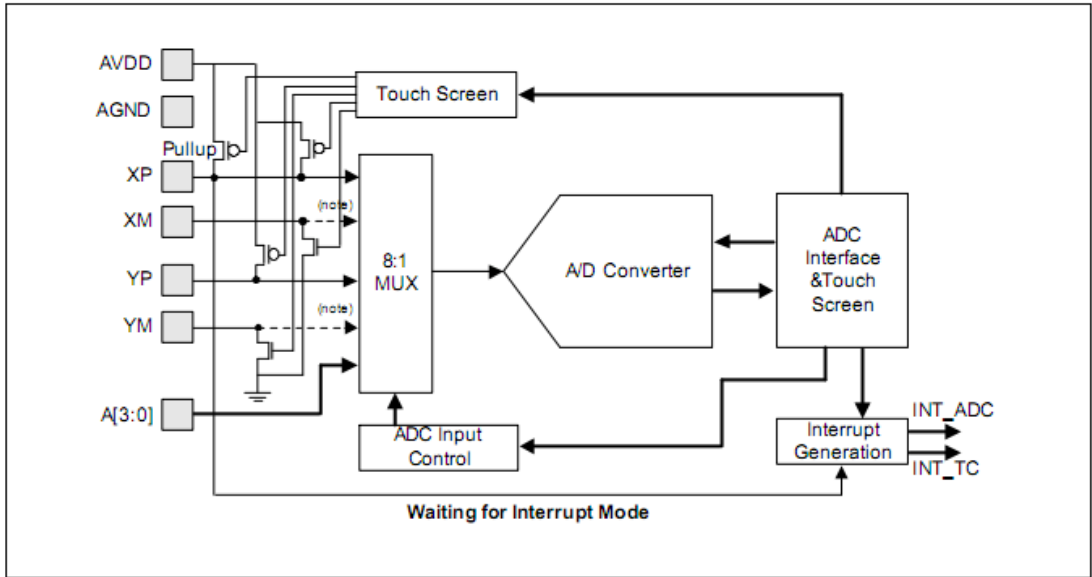
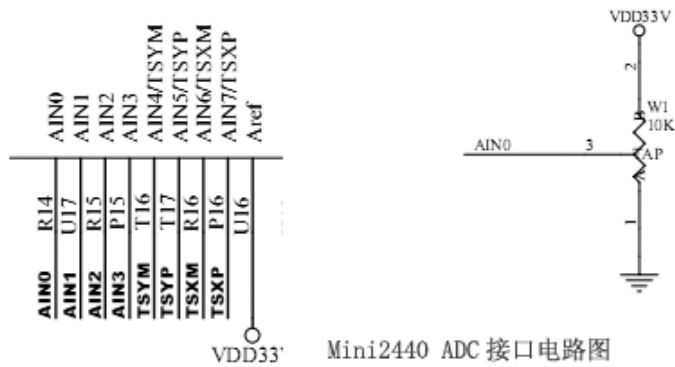


Figure 16-1. ADC and Touch Screen Interface Functional Block Diagram

图 4-1 S3C2440 模数转换器结构图



Mini2440 ADC 接口电路图

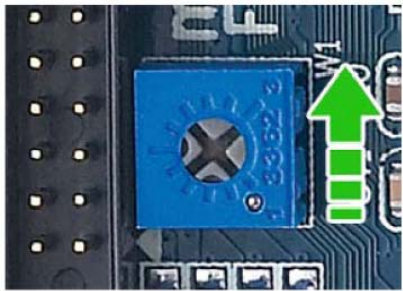


图 4-2 Mini2440 实验板模数转换器接口与滑动变阻器

本实验使用通道 0 采集实验板上的滑动变阻器的分压电压值，见图 4-2。

(2)启动 Keil uVision，新建一个工程 ex04-2。不需要系统提供的 Startup 文件。建立汇编源文件 ex04-2.s，编写实验程序，然后添加到工程中。设置工程选项，存储器映射。设置工程调试选项。建立仿真初始化文件 RAM.ini。

具体步骤参考实验二。

(3)实验程序实现了一个用于控制读取 ADC 指定通道的值的函数：

```
int ReadAdc(int ch);
```

在对 preScaler 变量初始化之后，即可通过调用该函数来多次采集连接在 ADC 上的模拟信号值。

(5) 编译链接工程。连接实验板电源、J-link 仿真器，进行仿真调试。单步运行程序，并改变板上滑动变阻器 W1 的阻值，观察采集到的电压数值的变化。

参考：模数转换器控制寄存器

ADC CONTROL REGISTER (ADCCON)

Register	Address	R/W	Description	Reset Value
ADCCON	0x5800000	R/W	ADC Control Register	0x3FC4

ADCCON	Bit	Description	Initial State
ECFLG	[15]	End of conversion flag(Read only) 0 = A/D conversion in process 1 = End of A/D conversion	0
PRSCEN	[14]	A/D converter prescaler enable 0 = Disable 1 = Enable	0
PRSCVL	[13:6]	A/D converter prescaler value Data value: 0 ~ 255 NOTE: ADC Frequency should be set less than PCLK by 5times. (Ex. PCLK=10MHZ, ADC Freq.< 2MHz)	0xFF
SEL_MUX	[5:3]	Analog input channel select 000 = AIN 0 001 = AIN 1 010 = AIN 2 011 = AIN 3 100 = YM 101 = YP 110 = XM 111 = XP	0
STDEM	[2]	Standby mode select 0 = Normal operation mode 1 = Standby mode	1
READ_START	[1]	A/D conversion start by read 0 = Disable start by read operation 1 = Enable start by read operation	0
ENABLE_START	[0]	A/D conversion starts by enable. If READ_START is enabled, this value is not valid. 0 = No operation 1 = A/D conversion starts and this bit is cleared after the start-up.	0

ADC CONVERSION DATA REGISTER (ADC DAT0)

Register	Address	R/W	Description	Reset Value
ADC DAT0	0x580000C	R	ADC Conversion Data Register	-

XPDATA (Normal ADC)	[9:0]	X-Position Conversion data value (include Normal ADC Conversion data value) Data value : 0 ~ 3FF	-
------------------------	-------	--	---

6. 实验参考程序

C 语言实验的参考程序见程序清单 4.1、4.2。模数转换器实验的参考程序见程序清单 4.3。

程序清单 4.1 C 语言实验参考程序（汇编部分）

```
Stack_Top    EQU    0x1000 - 4                ;end of ram
              AREA RESET, CODE, READONLY      ;声明代码段 RESET
              ENTRY                             ;表示程序入口
              CODE32                             ;声明 32 位 ARM 指令

              b      Reset

Undef        b      Undef                      ;handler for Undefined mode
SWI          b      SWI                        ;handler for SWI interrupt
Pabort       b      Pabort                     ;handler for PAbort
Dabort       b      Dabort                     ;handler for DAbort
              b      .                         ;reserved
IRQ          b      IRQ                       ;handler for IRQ interrupt
FIQ          b      FIQ                       ;handler for FIQ interrupt

Reset
              ;initialize stack pointer
              LDR     R0, =Stack_Top
              MOV     SP, R0

              ;
              ;
              IMPORT  main
              LDR     R0, =main
              BX      R0

              b      .                         ;should never get here
              END
```

程序清单 4.2 C 语言实验参考程序（C 语言部分）

```
int compare( int a, int b)
{
    int tmp, ret;

    tmp = a - b;
    if( tmp > 0)
        ret = 1;
    else if( tmp == 0)
        ret = 0;
    else
        ret = -1;
}
```

```

        return ret;
    }

    int max( int a, int b)
    {
        int tmp1;

        if( compare( a, b) >= 0)
            tmp1 = a;
        else
            tmp1 = b;
        return tmp1;
    }

    main()
    {
        int v = 0;

        v = max( v, 4);
        while(1)
        {
        }
    }

```

程序清单 4.3 模数转换器实验参考程序（C 语言部分）

```

#define rADCCON    (*(volatile unsigned *)0x58000000)    //ADC control
#define rADCTSC    (*(volatile unsigned *)0x58000004)    //ADC touch screen control
#define rADCPLY    (*(volatile unsigned *)0x58000008)    //ADC start or Interval Delay
#define rADCDAO    (*(volatile unsigned *)0x5800000c)    //ADC conversion data 0
#define rADCDA1    (*(volatile unsigned *)0x58000010)    //ADC conversion data 1
#define rADCUPTDN  (*(volatile unsigned *)0x58000014)    //Stylus Up/Down interrupt status

#define LOOP 10000
#define ADC_FREQ 1000000

volatile unsigned preScaler;

int ReadAdc(int ch)
{
    int i;
    static int prevCh=-1;

    rADCCON = (1<<14) | (preScaler<<6) | (ch<<3); //setup channel

    if(prevCh!=ch)

```

```

{
    rADCCON = (1<<14)|(preScaler<<6)|(ch<<3); //setup channel
    for(i=0;i<LOOP;i++); //delay to set up the next channel
    prevCh=ch;
}

rADCCON|=0x1; //start ADC

while(rADCCON & 0x1); //check if Enable_start is low
while(!(rADCCON & 0x8000)); //check if EC(End of Conversion) flag is high

return ( (int)rADCDATO & 0x3ff );
}

main()
{
    int v = 0;
    preScaler = 12000000/ADC_FREQ - 1; //PCLK:12MHz

    while(1)
    {
        v = ReadAdc(0); //对应开发板上 W1 可调电阻;
    }
}

```

7. 思考

- (1) 实验程序中 main（）函数名是否可以改为其它名字？
- (2) 如何校准模数转换器测量电压的精度？

8. 选作内容

- (1) 编程并观察 C 语言语句编译后对应的 ARM 汇编程序。
- (2) 用板上滑动变阻器控制指示灯的闪烁速度。