# 杭电单片机实验报告一

来源：网络 作者：月落乌啼 更新时间：2024-07-15

*单片机原理与应用技术实验报告实验题目：Keil软件使用及汇编语言编程姓名：\*\*\*学号：\*\*\*\*\*\*\*\*实验日期：2024指导教师：1.Keil软件的简介和使用1.1KeiluVision4的简介最新的KeiluVision4IDE，旨在提高...*

单片机原理与应用技术实验报告

实验题目：

Keil软件使用及汇编语言编程

姓名：

\*

\*

\*

学号：

\*\*\*\*\*\*\*\*

实验日期：

2024

指导教师：

1.Keil软件的简介和使用

1.1

Keil

uVision4的简介

最新的Keil

uVision4

IDE，旨在提高开发人员的生产力，实现更快，更有效的程序开发。

uVision4引入了灵活的窗口管理系统，能够拖放到视图内的任何地方，包括支持多显示器窗口。

uVision4在μVision3

IDE的基础上，增加了更多大众化的功能：

·

多显示器和灵活的窗口管理系统

·

系统浏览器窗口的显示设备外设寄存器信息

·

调试还原视图创建并保存多个调试窗口布局

·

多项目工作区简化与众多的项目

1.2

Keil

uVision4的使用

①，打开Keil

4应用软件，其初始界面如下：

②、点击菜单项“Project”下的“New

uVisionProject”，如下图所示。

③、选择文件夹，并输入工程名。

④、输入工程名并保存后会弹出选择芯片界面

⑤、这里我们选择Atmel公司的AT89S51单片机，选择后，点击“OK”，即创建完工程。

⑥、单击点击下图所示圆圈圈起的部分建立一个文件。也可以点击菜单“File”下面的“NEW”来新建一个文件。

⑦、之后输入程序的代码后，单击File—Save

as，根据代码语言的格类型，可保存为后缀为.asm（汇编语言）和.c（C语言）的文件。

⑧、单击“Target1”前面的“+”号，并用鼠标右键点击“SourceGroup1”，界面如图10所示。在用鼠标左键点击“Add

Files

to

Group

Source

Group1...”即可把刚刚的代码文件添加到工程。最后点击左上角的编译按钮即可完成编译。之后即可进行程序的调试了。

2.实验内容

使用汇编语言完成课本P95的（2）、（5）、（8）三道题以及一个跑马灯程序。

2.1

P95-（2）

设内部RAM

5AH单元中有一个变量X，请编写计算下述函数式的程序，结果存入5BH。

设计思路：显然本题要使用分支语句，不妨使用JZ,JNB等，并通过X-10，X-15综合判断X处于哪个区间。

代码如下：

ORG

0000H

JMP

START

ORG

1000H

START:

MOV

A,5AH

;直接寻址，将5AH地址上的数据X存入寄存器A

MOV

B,5AH

;直接寻址，将5AH地址上的数据X再一次存入寄存器B

SUBB

A,#0FH

;将A中数据X先减去15，初步判断X所属区间

JNB

ACC.7,DO3

;若A的第7位不为1（X>=15）则程序跳到DO3，否则顺序执行

MOV

A,5AH

;若X比15小，再次将X存入累加器A中

CLR

C

;清零进位C

SUBB

A,#0AH

;将X减去10

JNB

ACC.7,DO1

;若A的第7位不为1，（X>=10），跳到DO1

JMP

DO2

;剩下的情况就是：X<10，跳到DO2

DO1:MOV

A,5AH

;10<=X<=15

MUL

AB

;相当于X\*X，结果的低八位存入A

ADD

A,#08H

;即X\*X+8→A

MOV

5BH,A

;即X\*X+8→B

JMP

DONE

;跳到程序结束段

DO2:MOV

A,5AH

;X<10

MUL

AB

;X\*X→A

CLR

C

;进位C清零

SUBB

A,#01H

;

X\*X-1→A

MOV

5BH,A

;

X\*X-1→B

JMP

DONE

;跳到程序结束段

DO3:MOV

A,#29H

;41→A

MOV

5BH,A

;41→B

JMP

DONE

;跳到程序结束段

DONE:SJMP

$

END

程序截图：

程序结果：

①、当X=12时，Y=12\*12+8=152

②、当X=8时，Y=8\*8-1=63

③、当X=18时，Y=41

2.2

P95-（5）

设有两个长度均为15的数组，分别存放在以2000H和2100H为首的存储区中，试编程求其对应项之和，结果存放到以2200H为首的存储区中。

设计思路：本题需要用到LOOP进行15次的求和及存放。并且由于地址过高，应采用DPTR进行访问。

代码如下：

ORG

0000H

JMP

START

ORG

1000H

START:

MOV

R1,#00H

;求和寄存器，初始清零

MOV

R2,#0FH

;循环次数为15次

MOV

DPL,#00H

;类似段内偏移地址地址

LOOP:

MOV

DPH,#20H

;DPTR的高八位，可理解为段地址，这里先指向第一个数组的首

;地址2000H

MOVX

A,@DPTR

;A←((DPTR))

MOV

R1,A

;R1←(A)

INC

DPH

;DPH改成指向高8位是21H的存储区

MOVX

A,@DPTR

;A←((DPTR))

ADD

A,R1

;两数组对应项求和

INC

DPH

;DPH改成指向高8位是22H的存储区

MOVX

@DPTR,A

;求和结果送入上述地址区

INC

DPL

;低8位地址自增

DJNZ

R2,LOOP

;若循环次数-1后不为0，则跳转到LOOP处进行循环

END

程序截图：

程序结果：

首先分别在2000H和2100H存入15数，这里我都存入了1,2,3,…,14,15。

结果得到：

2.3

P95-（8）

将片外数据存储器地址为1000H~1030H的数据块，全部搬迁到片内RAM

30H~60H中，并将原数据块区域全部清零。

设计思路：显然还是得用LOOP进行搬迁，且每搬完一个就得清零原地址数据，并用DPTR访问片外存储器。

代码如下：

ORG

0000H

JMP

START

ORG

1000H

START:

MOV

R1,#30

;用R1保存循环次数

30次

MOV

R0,#30H

;将地址30H存入R0中

MOV

DPTR,#1000H

;将1000H这个地址存入DPTR中

LOOP:

MOVX

A,@DPTR

;寄存器间接寻址。A←((DPTR))

MOV

@R0,A

;(R0)←A

CLR

A

;A←0

MOVX

@DPTR,A

;原地址数据清零

INC

R0

;数据的写入地址

自增

INC

DPTR

;数据的读取地址

自增

DJNZ

R1,LOOP

;若循环次数-1不为0，则跳转到LOOP处进行循环

END

程序截图：

程序结果：

先在片外存储器1000H开始存入数据

接下来开始将数据搬迁到片内30H开始的区域中

且原地址数据依次被清零

2.4

P95-（8）

题目：设计一个6+50=56ms跑马灯，并在P1端口显示。

设计思路：首先我得设计一个56ms的延迟。这里已知当单片机系统的振荡频率为fosc=12MHz时，一个机器周期为1T=1us；执行一条DJNZ指令需要2个机器周期，执行一条NOP需要一个机器周期，执行一条MOV也是一个机器周期。

接着，再通过循环左移指令RL

A实现跑马等效果。

代码如下：

ORG

0000H

JMP

START

ORG

0800H

START:MOV

A,#01H

;即将0000

0001存入A，后面会将最右边1循环左移。

SJMP

DELAY

;进入延时段起点

LOOP:

RL

A

;从此处开始跑马灯模块。先将A循环左移

MOV

P1,A

;将A的值赋给端口P1的8位

SJMP

DELAY

;从56ms的延迟程序段再跑一遍

DELAY:

MOV

R7,#224

;设置外循环次数224次

DLY1:

MOV

R6,#123

;设置内循环次数123次

DLY2:

DJNZ

R6,DLY2

NOP

DJNZ

R7,DLY1

;延时程序段结束，总用时(123\*2+2+1+1)T\*224T+2T+1T=56ms

SJMP

LOOP

SJMP

END

程序截图：

程序结果：

初始状态

执行一次56ms延迟后左移

再一次56ms延迟后左移

Proteus仿真截图：

3.实验心得

通过以上4个实验，我学会了KEIL软件的使用，即创建工程，创建并添加项目，编译与调试等等。更重要的是学习了很多单片机编程方面的只是。比如用汇编语言实现数据的传送、运算、移位等，同时还有一些简单的程序设计，如分支程序、循环程序等等。

这些实验使我更好的从底层了解单片机系统的运行机制，尤其的汇编语言的使用更是提高了我对各类端口、存储器的运用水平。另一方面我也感受到汇编语言虽然十分直接，但面对大型的程序项目就显得力不从心，所以这就体现了后期实验使用C语言编写的简便性。

我想，通过更多的运用各类指令及对更多端口、元件的综合使用，我会对变得越来越熟练，且能不断地提升自己的逻辑思维。

本DOCX文档由 www.zciku.com/中词库网 生成，海量范文文档任你选，，为你的工作锦上添花,祝你一臂之力！