

# AD与DA



专 业：生物医学工程

年 级：2022级

班 级：1班

姓 名：张台忍

学 号：3022202299

邮 箱：ztr8526@gmail.com

**2024年5月12日**

Contents

[AD与DA 1](#_Toc166527838)

[一 实验目的 3](#_Toc166527839)

[二 实验设备 3](#_Toc166527840)

[三 实验内容 3](#_Toc166527841)

[1 建立工程 3](#_Toc166527842)

[2 需求分析 3](#_Toc166527843)

[3 硬件电路搭建 4](#_Toc166527844)

[4 的配置 4](#_Toc166527845)

[5 代码编写 6](#_Toc166527846)

[6 调试记录 7](#_Toc166527847)

[7 结果与讨论 7](#_Toc166527848)

# 一 实验目的

1. 学习与原理与应用
2. 掌握总线原理与应用
3. 学习显示模块的原理与应用
4. 掌握的原理

# 二 实验设备

微机一台, 集成开发环境一套, 单片机开发仪

# 三 实验内容

本实验流程图

A diagram of a company

Description automatically generated

图 0 流程图

## 1 建立工程

在中, 参照实验的方法, 建立工程, 然后新建语言程序, 编译通过后进行接下来的操作.

## 2 需求分析

本次实验有两个需求, 描述如下:

1. 编写一段程序, 通过芯片读取光敏电阻, 热敏电阻和电位器的转换值, 并在上实时刷新显示.
2. 根据测得的电位器值调节灯亮度(利用).

对于需求, 可以通过配置外部芯片, 读取电阻数值, 转换成数字量让单片机读取, 读取之后通过的显示函数显示在屏幕上.

紧接着需求, 可以将需求中接收到的数据存起来, 然后使用定时器中断, 输出波, 将输出接到芯片上, 实现呼吸灯效果.

## 3 硬件电路搭建

参考原理图连接硬件电路, 连接图如下.



图1 硬件电路连接

本次实验硬件电路简单, 只需要把插上, 将模块的排针与单片机口相连即可.

## 4 的配置

本次实验需要用到转换芯片, 因此要先对该芯片进行配置, 才能使用功能. 接下来将对进行配置.

首先查阅数据手册, 观察时序图. 时序图如下:

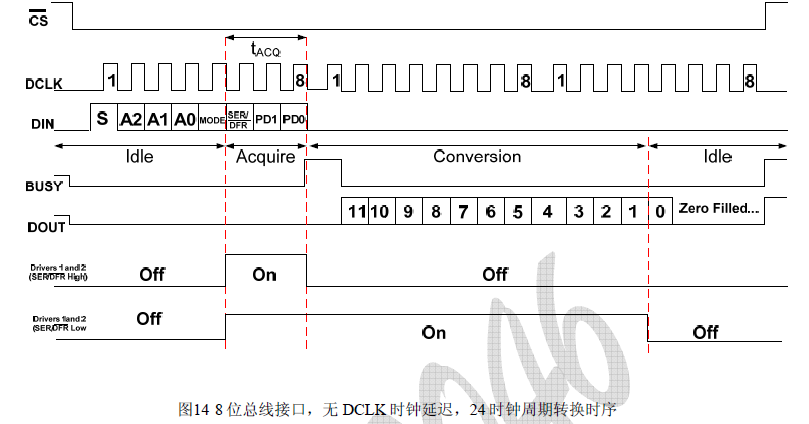


图2 时序图

观察时序图可知, 配置时, 应先将置, 再将置, 即将置. 这样就完成了的初始化设置, 之后就可以向里面输入数据, 来发送数据了.

以下是配置的代码实现.

1. #include <REGX52.H>
2. #include <INTRINS.H>
4. //引脚定义
5. sbit XPY2046\_DIN=P1^2;
6. sbit XPY2046\_CS=P1^1;
7. sbit XPY2046\_DCLK=P1^0;
8. sbit XPY2046\_DOUT=P1^3;
10. /\*\*
11. \* @brief  XPT2046读取AD值
12. \* @param  Command 命令字，范围：头文件内定义的宏，结尾的数字表示转换的位数
13. \* @retval AD转换后的数字量，范围：8位为0~255，12位为0~4095
14. \*/
15. unsigned **int** XPT2046\_ReadAD(unsigned **char** Command)
16. {
17. unsigned **char** i;
18. unsigned **int** Data=0;
19. XPY2046\_DCLK=0;
20. XPY2046\_CS=0;
21. **for**(i=0;i<8;i++)
22. {
23. XPY2046\_DIN=Command&(0x80>>i);        //发送数据
24. XPY2046\_DCLK=1;
25. XPY2046\_DCLK=0;     //模拟时序信号
26. }
27. **for**(i=0;i<16;i++)
28. {
29. XPY2046\_DCLK=1;
30. XPY2046\_DCLK=0;
31. **if**(XPY2046\_DOUT){Data|=(0x8000>>i);}
32. }
33. XPY2046\_CS=1;
34. **return** Data>>8;
35. }

## 5 代码编写

完成的配置之后, 即可开始编写代码实现需求.

观察原理图可知, 电位器连接在的端口, 连接在端口, 连接在端口. 因此, 在读取数据时, 只需要读取这三个端口的数据, 并将数据显示在上即可实现需求1. 以下是代码实现.

1. #include <REGX51.H>
2. #include "LCD1602.h"
3. #include "XPT2046.h"
5. **void** main()
6. {
7. LCD\_Init();
8. **while**(1)
9. {
10. LCD\_ShowString(1,1,"GR1");
11. LCD\_ShowNum(2,1,XPT2046\_ReadAD(XPT2046\_VBAT),3);
13. LCD\_ShowString(1,5,"Pot");
14. LCD\_ShowNum(2,5,XPT2046\_ReadAD(XPT2046\_XP),3);
16. LCD\_ShowString(1,9,"NTC1");
17. LCD\_ShowNum(2,9,XPT2046\_ReadAD(XPT2046\_YP),3);
18. }
19. }

对于需求2, 核心部分是输出波, 这里的波的占空比通过读取到的电位器的数值来调整. 以下是实现的代码.

1. #include <REGX51.H>
2. #include "Delay.h"
3. #include "LCD1602.h"
4. #include "XPT2046.h"
5. #include "Timer0.h"
7. unsigned **char** Counter,Compare,i,num;
9. **void** main()
10. {
11. LCD\_Init();
12. Timer0\_Init();
13. **while**(1)
14. {
16. num=XPT2046\_ReadAD(XPT2046\_XP);
17. LCD\_ShowString(1,1,"Pot");
18. LCD\_ShowNum(2,1,num,3);
20. **for**(i;i<num;i++)
21. {
22. Compare=i;          //设置比较值，改变PWM占空比
23. Delay(10);
24. }
25. **for**(i=num;i>0;i--)
26. {
27. Compare=i;          //设置比较值，改变PWM占空比
28. Delay(10);
29. }
30. }
31. }
33. **void** Timer0\_Routine() interrupt 1
34. {
35. TL0 = 0x9C;             //设置定时初始值
36. TH0 = 0xFF;             //设置定时初始值
37. Counter++;
38. Counter%=255;   //计数值变化范围限制在0~99
39. **if**(Counter<Compare)  //计数值小于比较值
40. {
41. P2\_1=1;     //输出1
42. }
43. **else**                //计数值大于比较值
44. {
45. P2\_1=0;     //输出0
46. }
47. }

代码中通过读出的电位器阻值, 调整比较的限制, 进而调整了波的占空比, 再通过外接的实现了对呼吸灯频率的调整.

## 6 调试记录

调试过程中, 发现显示屏中显示的数据一直为, 经过检查, 代码没有问题, 后发现是杜邦线连接松动, 加固连接后解决了该问题.

实现需求2时, 无法实时刷新显示的数值, 这是由于产生波的代码执行需要时间.

## 7 结果与讨论

实验结果现象如图所示.

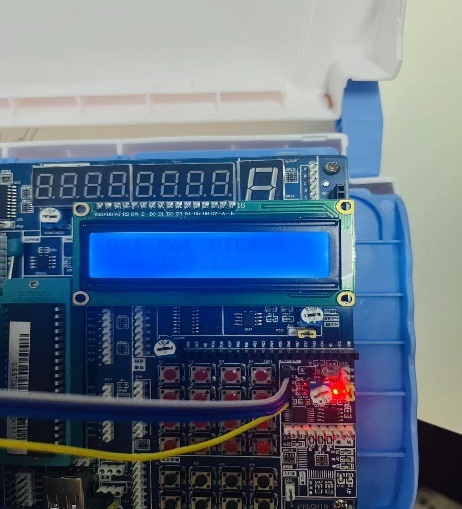


图3 需求1现象

A close up of a circuit board

Description automatically generated

图4 需求2现象

本次实验实现了和的使用. 将接收到的数据显示在屏幕上. 并通过波的形式发送数据经实现呼吸灯. 进一步加深了对的理解.