****

A级达标线上测试报告



**学院** 网络与信息安全学院 **专业**  网络工程

**学号**

**姓名**

**手机** **完成日期** 2024-09-23

**成绩**

# 题目名称: 计算器设计

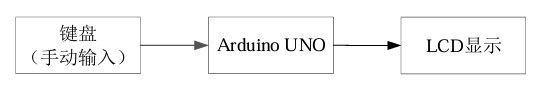
一、题目要求

实现一个计算器的设计，具体要求：

**软件环境：**推荐采用Proteus 8.9 SP2及以上仿真软件，Arduino IDE。

**实现功能：**使用Arduino UNO微控制器，实现一个简易计算器。

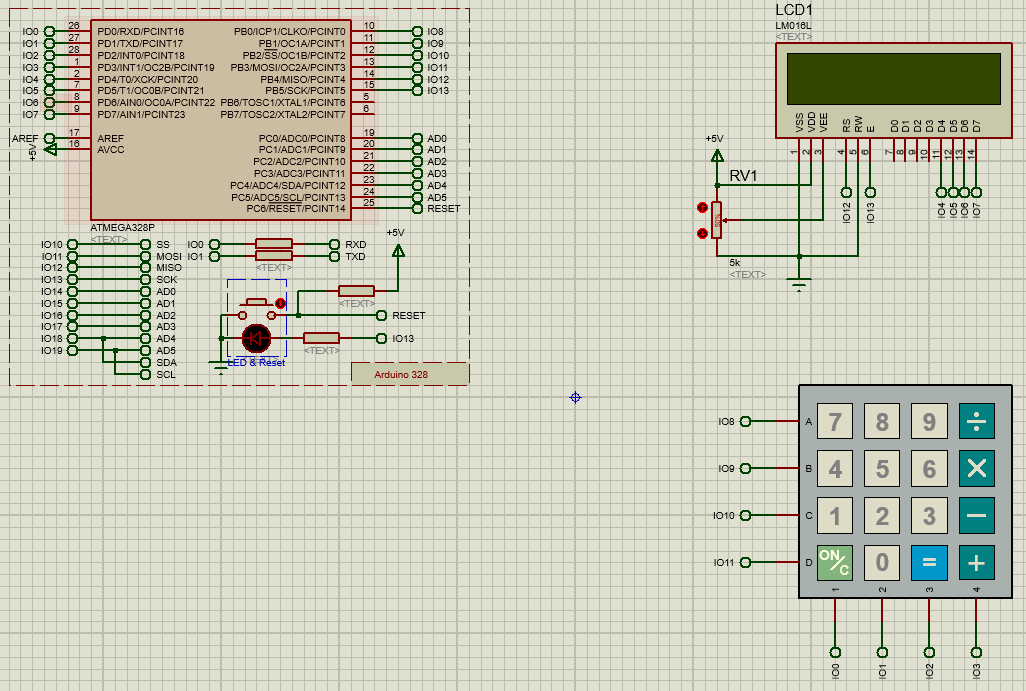
**功能：**简易计算器通过键盘输入整数和计算符号，Arduino UNO进行处理，在LCD上显示输入的整数、计算符号以及计算结果。



**样例：**LCD第一行显示自己学号最左端第1、2位组成的两位数\*学号第5、6位组成的两位数-学号第9、10、11位组成的三位数= LCD第二行显示计算结果。

对我来说，我的学号为，因此要计算。

二、设计思路



如图所示，proteus的项目在原电路图的基础上，我针对LCD和键盘的引脚进行了连接。其中LCD1的RS连接到IO12、E连接到IO13、D4-7连接到IO4-7。对于键盘的连接，对于键盘的行A-D，连接到IO8-11；对于键盘的列1-4，连接到IO0-3。

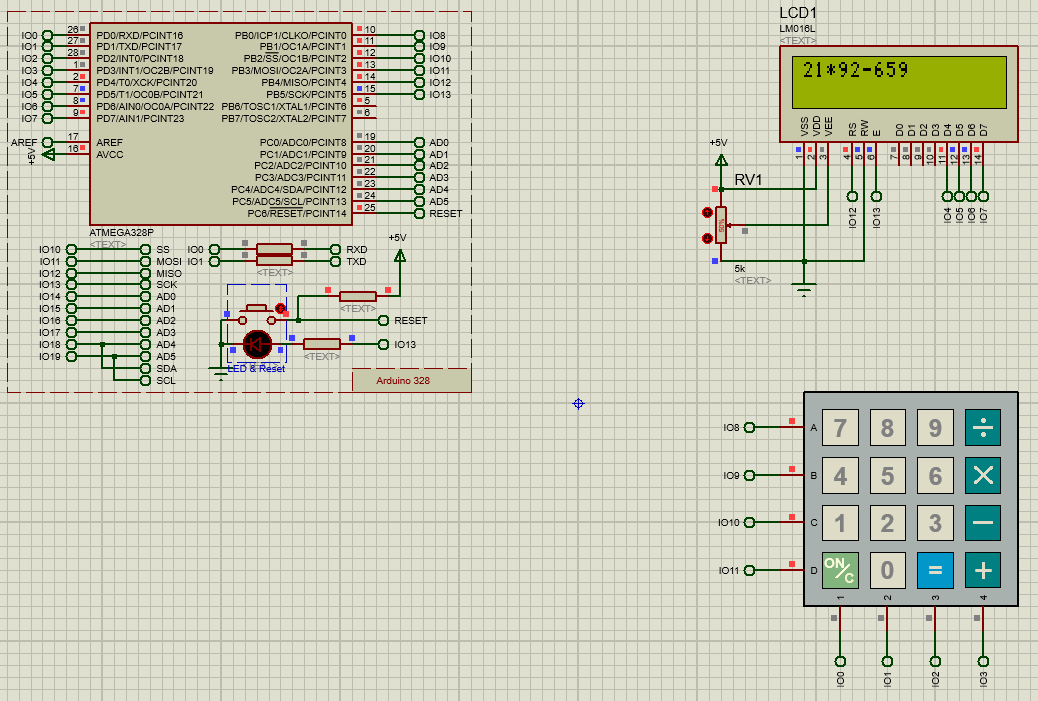
通过上述连接，可以保证Arduino UNO顺利接收键盘输入，并将内容输出到LCD上，完成目标任务。

完成引脚分配后，即可在Arduino IDE中完成对应的代码编写。通过LCD引脚初始化、定义键盘布局、读取键盘输入、计算并输出在LCD上，完成了对应的任务。需要指出，我的仿真仅满足了对于学号的乘减计算。

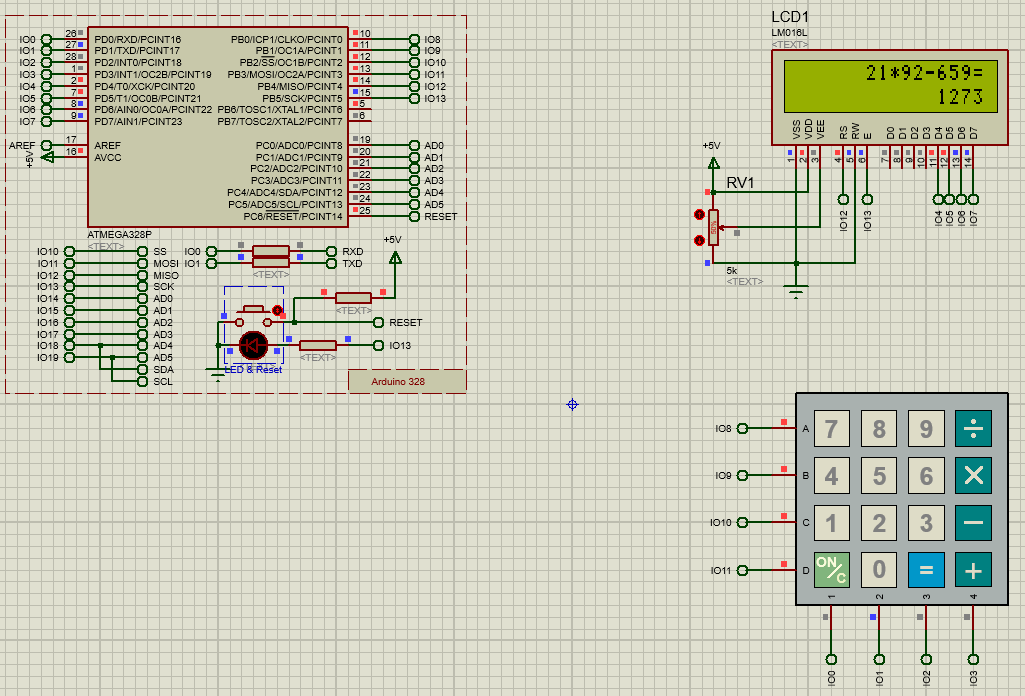
之后对完成的代码，在Arduino中进行编译，即左上角的**√**，之后下方显示编译完成，前往对应的程序所在地即可获得对应的.hex文件，通过这个文件即可完成仿真。



三、仿真结果展示



如图所示，我的学号为21009200659，因此需要计算21 \* 92 – 659，因此通过键盘，我首先将需要计算的内容输入，计算内容也输出在了LCD上。



之后按下“=”，我的仿真中顺利计算出了结果1273，并将计算式子和计算结果分两行、右对齐输出在了LCD上。据此仿真成功。

四、参考文献

[1] Banzi, M., & Shiloh, M. (2014). Getting Started with Arduino: The Open Source Electronics Prototyping Platform (3rd ed.). O'Reilly Media.

[2] Rajesh, S., & Babu, R. (2017). Design and Implementation of 4x4 Keypad Matrix with Embedded System and Arduino. International Journal of Recent Technology and Engineering (IJRTE), 6(4), 55-59.

[3] SparkFun Electronics. (n.d.). How to Use a 4x4 Keypad with Arduino. SparkFun.com. Retrieved from https://learn.sparkfun.com/tutorials/4x4-keypad-hookup-guide

[4] weixin\_58243145. (2023, July 17). 2023西电A测：湿度测控仿真系统. CSDN. https://blog.csdn.net/weixin\_58243145/article/details/133042452

五、程序设计

1. #include <Keypad.h>
2. #include <LiquidCrystal.h>
3. *// LCD引脚初始化*
4. LiquidCrystal lcd(12, 13, 4, 5, 6, 7);  *// RS, EN, D4, D5, D6, D7引脚*
5. *// 定义键盘布局*
6. const byte ROWS = 4;  *// 4行*
7. const byte COLS = 4;  *// 4列*
8. char keys[ROWS][COLS] = {
9. {'7', '8', '9', '/'},
10. {'4', '5', '6', '\*'},
11. {'1', '2', '3', '-'},
12. {'C', '0', '=', '+'}
13. };
14. byte rowPins[ROWS] = {8, 9, 10, 11};  *// 行引脚：IO8-IO11*
15. byte colPins[COLS] = {0, 1, 2, 3};    *// 列引脚：IO0-IO3*
16. *// 键盘初始化*
17. Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);
18. *// 全局变量*
19. String inputString = "";     *// 保存用户输入的完整字符串*
20. String firstNumber = "";     *// 保存第一个两位数*
21. String secondNumber = "";    *// 保存第二个两位数*
22. String thirdNumber = "";     *// 保存第三个三位数*
23. char operator1 = ' ';        *// 第一个运算符（\*）*
24. char operator2 = ' ';        *// 第二个运算符（-）*
25. bool operator1Selected = false;  *// 用于判断是否已经选择了第一个运算符*
26. bool operator2Selected = false;  *// 用于判断是否已经选择了第二个运算符*
27. int result = 0;              *// 保存计算结果*
28. void setup() {
29. lcd.begin(16, 2);  *// 初始化LCD*
30. lcd.clear();
31. lcd.setCursor(0, 0);
32. lcd.print("Calculator Ready");  *// 初始化文字*
33. delay(1000);  *// 延迟显示*
34. lcd.clear();  *// 清除LCD内容*
35. }
36. void loop() {
37. char key = keypad.getKey();  *// 获取键盘输入*
38. if (key) {
39. if (key == 'C') {
40. *// 清除所有输入并重置LCD*
41. inputString = "";
42. firstNumber = "";
43. secondNumber = "";
44. thirdNumber = "";
45. operator1 = ' ';
46. operator2 = ' ';
47. operator1Selected = false;
48. operator2Selected = false;
49. lcd.clear();
50. }
51. else if (key == '=') {
52. if (firstNumber != "" && secondNumber != "" && thirdNumber != "" && operator1 != ' ' && operator2 != ' ') {
53. *// 将输入的数字字符串转换为整数并执行运算*
54. int num1 = firstNumber.toInt();  *// 转换第一个两位数为整数*
55. int num2 = secondNumber.toInt(); *// 转换第二个两位数为整数*
56. int num3 = thirdNumber.toInt();  *// 转换第三个三位数为整数*
57. *// 先计算 num1 \* num2，然后再减去 num3*
58. result = (num1 \* num2) - num3;
59. *// 在LCD的第一行显示完整的输入内容（右对齐）*
60. String expression = firstNumber + operator1 + secondNumber + operator2 + thirdNumber + "=";
61. lcd.clear();
62. lcd.setCursor(16 - expression.length(), 0);  *// 右对齐显示表达式*
63. lcd.print(expression);
64. *// 在LCD的第二行显示计算结果（右对齐）*
65. lcd.setCursor(16 - String(result).length(), 1);  *// 右对齐显示结果*
66. lcd.print(result);
67. }
68. }
69. else if (key >= '0' && key <= '9') {
70. *// 处理数字输入*
71. if (!operator1Selected) {
72. *// 第一个数字输入阶段（最多两位数）*
73. if (firstNumber.length() < 2) {
74. firstNumber += key;
75. inputString += key;
76. }
77. } else if (!operator2Selected) {
78. *// 第二个数字输入阶段（最多两位数）*
79. if (secondNumber.length() < 2) {
80. secondNumber += key;
81. inputString += key;
82. }
83. } else {
84. *// 第三个数字输入阶段（最多三位数）*
85. if (thirdNumber.length() < 3) {
86. thirdNumber += key;
87. inputString += key;
88. }
89. }
90. lcd.setCursor(0, 0);
91. lcd.print(inputString);  *// 实时显示用户输入*
92. }
93. else if (key == '\*' || key == '/' || key == '+' || key == '-') {
94. *// 处理运算符输入*
95. if (!operator1Selected && firstNumber != "") {
96. operator1 = key;
97. inputString += key;
98. operator1Selected = true;
99. lcd.setCursor(0, 0);
100. lcd.print(inputString);  *// 实时显示运算符*
101. } else if (operator1Selected && secondNumber != "" && !operator2Selected) {
102. operator2 = key;
103. inputString += key;
104. operator2Selected = true;
105. lcd.setCursor(0, 0);
106. lcd.print(inputString);  *// 实时显示第二个运算符*
107. }
108. }
109. }
110. }