Lesson 18 LCD Display

Introduction

In this lesson, you will learn how to connect LCD Display to Arduino Mega 2560 and display what we type.

In addition, with the Potentiometer we can control the brightness of the screen.

Hardware Required

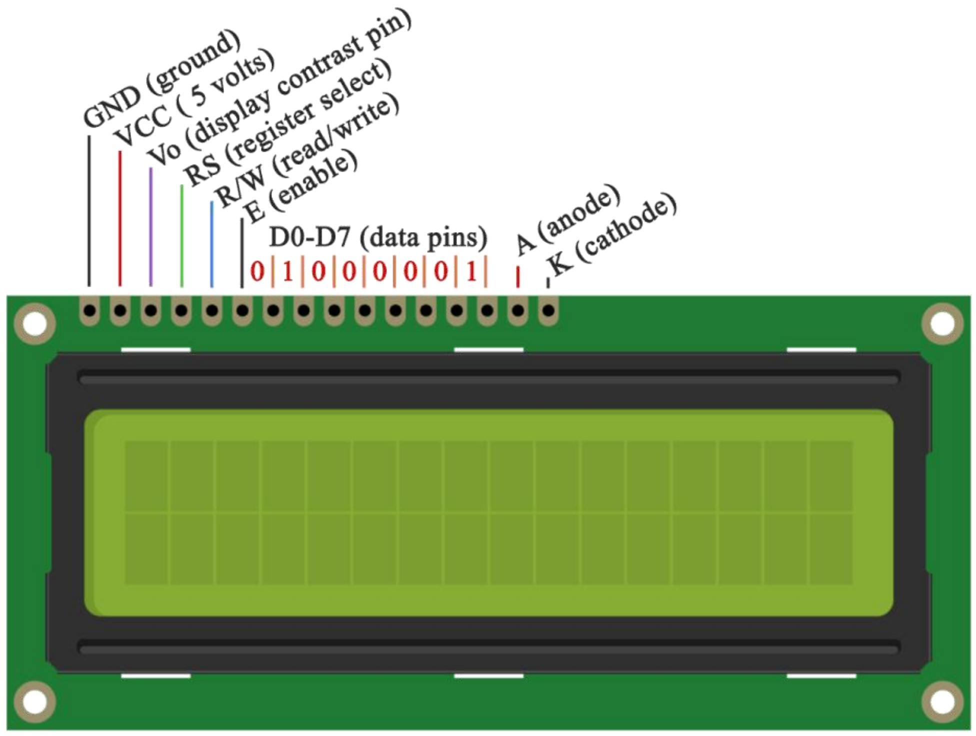
- ✓ 1 * RexQualis Mega 2560
- ✓ 1 * LCD1602 module
- ✓ 1 * Potentiometer (10k)
- ✓ 1 * Breadboard
- ✓ 16* M-M Jumper Wires

Principle

LCD1602

The lcd display has 16 pins and the first one from left to right is the Ground pin. The second pin is the VCC which we connect the 5 volts pin on the Arduino Board. Next is the Vo pin on which we can attach a potentiometer for controlling the contrast of the display.

Next, The RS pin or register select pin is used for selecting whether we will send commands or data to the LCD. For example if the RS pin is set on low state or zero volts, then we are sending commands to the LCD like: set the cursor to a specific location, clear the display, turn off the display and so on. And when RS pin is set on High state or 5 volts we are sending data or characters to the LCD.



Code interpretation

//LCD RS pin to digital pin 12

//LCD Enable pin to digital pin 2

//LCD D0 pin to digital pin 3

//LCD D1 pin to digital pin 4

//LCD D2 pin to digital pin 5

//LCD D3 pin to digital pin 6

//LCD D4 pin to digital pin 7

//LCD D5 pin to digital pin 8

//LCD D6 pin to digital pin 9

//LCD D7 pin to digital pin 10

//LCD R/W pin to digital pin 11

//LCD VSS pin to ground

```
//LCD VCC pin to 5V

//LCD K pin to ground

//LCD A pin to 5V

//LCD V0 pin to 10K resistor:

//ends to +5V and ground


int DI = 12;

int RW = 11;

int DB[] = {3, 4, 5, 6, 7, 8, 9, 10};//Use an array to define the pins

int Enable = 2;

void LcdCommandWrite(int value) {

 // Define all pins

 int i = 0;

 for (i=DB[0]; i <= DI; i++) //Assignment

{

    digitalWrite(i,value & 01);//Because 1602 LCD signal
identification is D7-D0 (not D0-D7), here is used to invert the
signal.

    value >>= 1;

 }

 digitalWrite(Enable,LOW);

 delayMicroseconds(1);

 digitalWrite(Enable,HIGH);
```

```
  delayMicroseconds(1);

  digitalWrite(Enable,LOW);

  delayMicroseconds(1);

}

void LcdDataWrite(int value) {

  // Define all pins

  int i = 0;

  digitalWrite(DI, HIGH);

  digitalWrite(RW, LOW);

  for (i=DB[0]; i <= DB[7]; i++) {

     digitalWrite(i,value & 01);

     value >>= 1;

  }

  digitalWrite(Enable,LOW);

  delayMicroseconds(1);

  digitalWrite(Enable,HIGH);

  delayMicroseconds(1);

  digitalWrite(Enable,LOW);

  delayMicroseconds(1);

  }

void setup (void) {

  int i = 0;
```

```
 for (i=Enable; i <= DI; i++) {
    pinMode(i,OUTPUT);
 }
 delay(100);
 // Initialize the LCD
 LcdCommandWrite(0x38);     // Set to 8-bit interface, 2 lines
display, 5x7 text size
 delay(64);
 LcdCommandWrite(0x38);     // Set to 8-bit interface, 2 lines
display, 5x7 text size
 delay(50);
 LcdCommandWrite(0x38);     // Set to 8-bit interface, 2 lines
display, 5x7 text size
 delay(20);
 LcdCommandWrite(0x06);  // Input method setting
                        // Auto increment, no shift is displayed
 delay(20);
 LcdCommandWrite(0x0E);  // display setting
                        // Turn on the display, the cursor
shows, no flicker
 delay(20);
 LcdCommandWrite(0x01);  // The screen is empty and the cursor
position is zeroed
 delay(100);
```

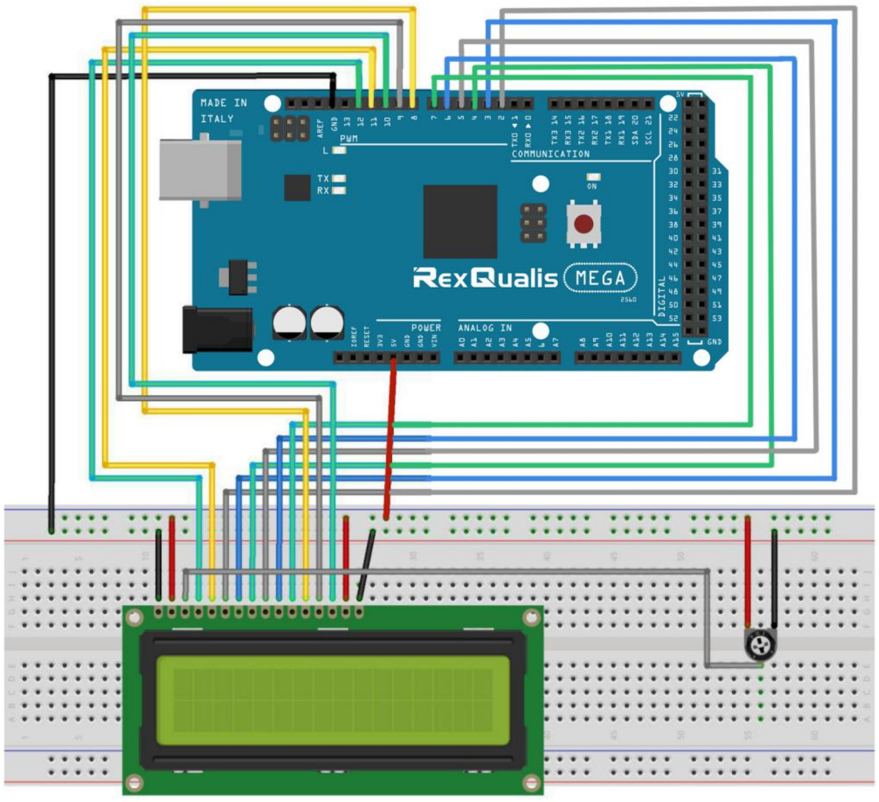
```
  LcdCommandWrite(0x80);   // display setting
  //Turn on the display, the cursor shows, no flicker
  delay(20);
}
void loop (void) {
   LcdCommandWrite(0x01);   // The screen is empty and the cursor
position is zeroed
   delay(10);
   LcdCommandWrite(0x80+3);
   delay(10);
   // Write information
   LcdDataWrite('W');
   LcdDataWrite('e');
   LcdDataWrite('l');
   LcdDataWrite('c');
   LcdDataWrite('o');
   LcdDataWrite('m');
   LcdDataWrite('e');
   LcdDataWrite(' ');
   LcdDataWrite('t');
   LcdDataWrite('o');
   delay(10);
   LcdCommandWrite(0xc0+3);   // Define the cursor position as the
```

```
third position of the second line
    delay(10);
    LcdDataWrite('R');
    LcdDataWrite('e');
    LcdDataWrite('x');
    LcdDataWrite('q');
    LcdDataWrite('u');
    LcdDataWrite('a');
    LcdDataWrite('l');
    LcdDataWrite('i');
    LcdDataWrite('s');
    delay(5000);
    LcdCommandWrite(0x01);    // The screen is empty and the cursor
position is zeroed
    delay(10);
    LcdCommandWrite(0x80+2);  //Define the cursor position as the
second position of the first line
    delay(10);
    LcdDataWrite('M');
    LcdDataWrite('a');
    LcdDataWrite('k');
    LcdDataWrite('e');
    LcdDataWrite(' ');
```

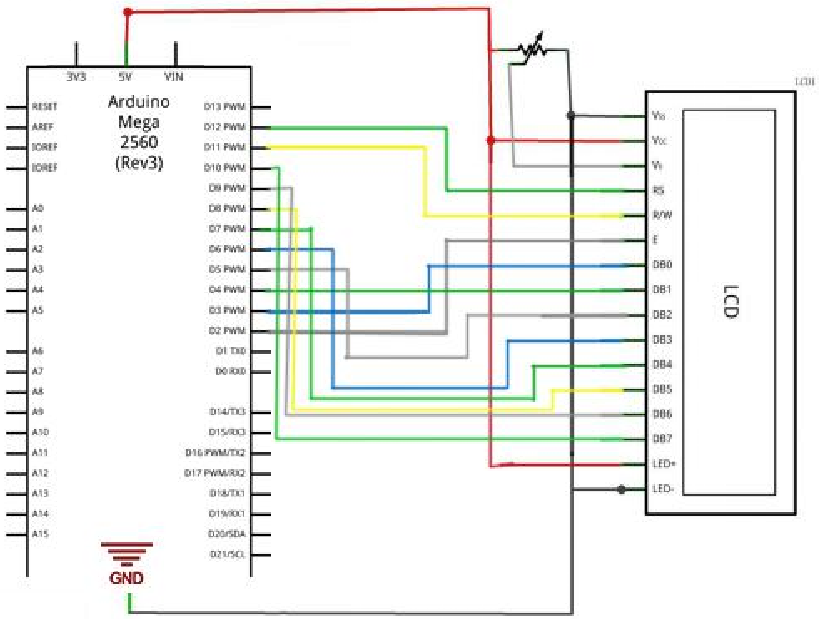
```
  LcdDataWrite('S');

  LcdDataWrite('c');

  LcdDataWrite('i');

  LcdDataWrite('e');

  LcdDataWrite('n');

  LcdDataWrite('c');

  LcdDataWrite('e');

  delay(10);

  LcdCommandWrite(0xc0+6);   // Define the cursor position as the
sixth position of the second line

  delay(10);

  LcdDataWrite('F');

  LcdDataWrite('u');

  LcdDataWrite('n');

  delay(5000);

  LcdCommandWrite(0x01);   // The screen is empty and the cursor
position is zeroed

  delay(10);

  LcdCommandWrite(0x80+2); //Define the cursor position as the
second position of the first line

  delay(10);

  LcdDataWrite('M');

  LcdDataWrite('a');
  LcdDataWrite('k');
  LcdDataWrite('e');
  LcdDataWrite(' ');
  LcdDataWrite('S');
  LcdDataWrite('c');
  LcdDataWrite('i');
  LcdDataWrite('e');
  LcdDataWrite('n');
  LcdDataWrite('c');
  LcdDataWrite('e');
  delay(10);
  LcdCommandWrite(0xc0+4);    // Define the cursor position as the
fourth position of the second line
  delay(10);
  LcdDataWrite('P');
  LcdDataWrite('o');
  LcdDataWrite('p');
  LcdDataWrite('u');
  LcdDataWrite('l');
  LcdDataWrite('a');
  LcdDataWrite('r');
  delay(5000);
  }
```

Experimental Procedures

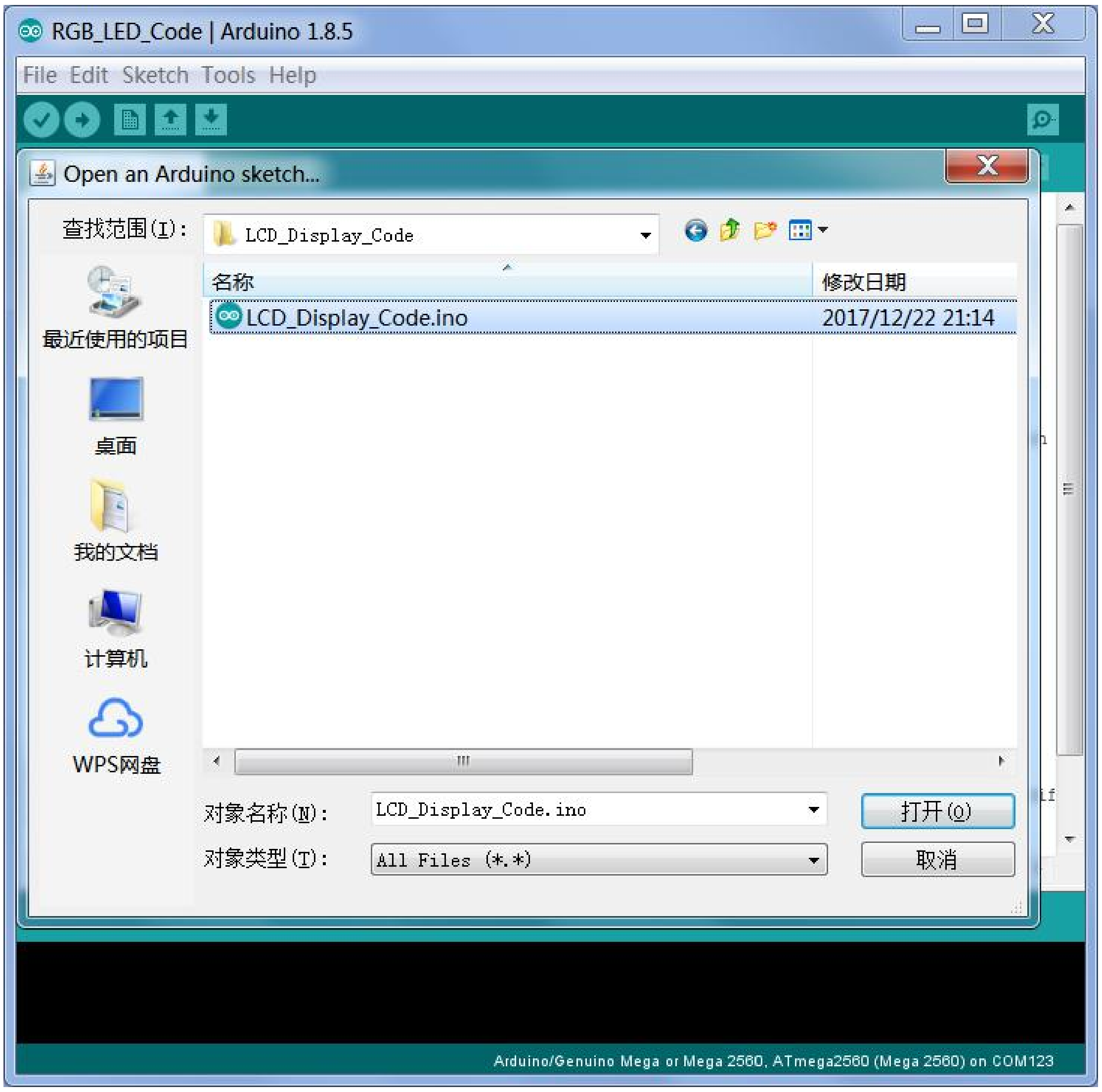
Step 1:Build the circui



Schematic Diagram



Step 2:Open the code:LCD_Display_Code


RGB_LED_Code | Arduino 1.8.5
File Edit Sketch Tools Help
Open an Arduino sketch...
查找范围(I): LCD_Display_Code
最近使用的项目
桌面
我的文档
计算机
WPS网盘
打开(O)
取消

名称	修改日期
LCD_Display_Code.ino	2017/12/22 21:14

对象名称(N): LCD_Display_Code.ino

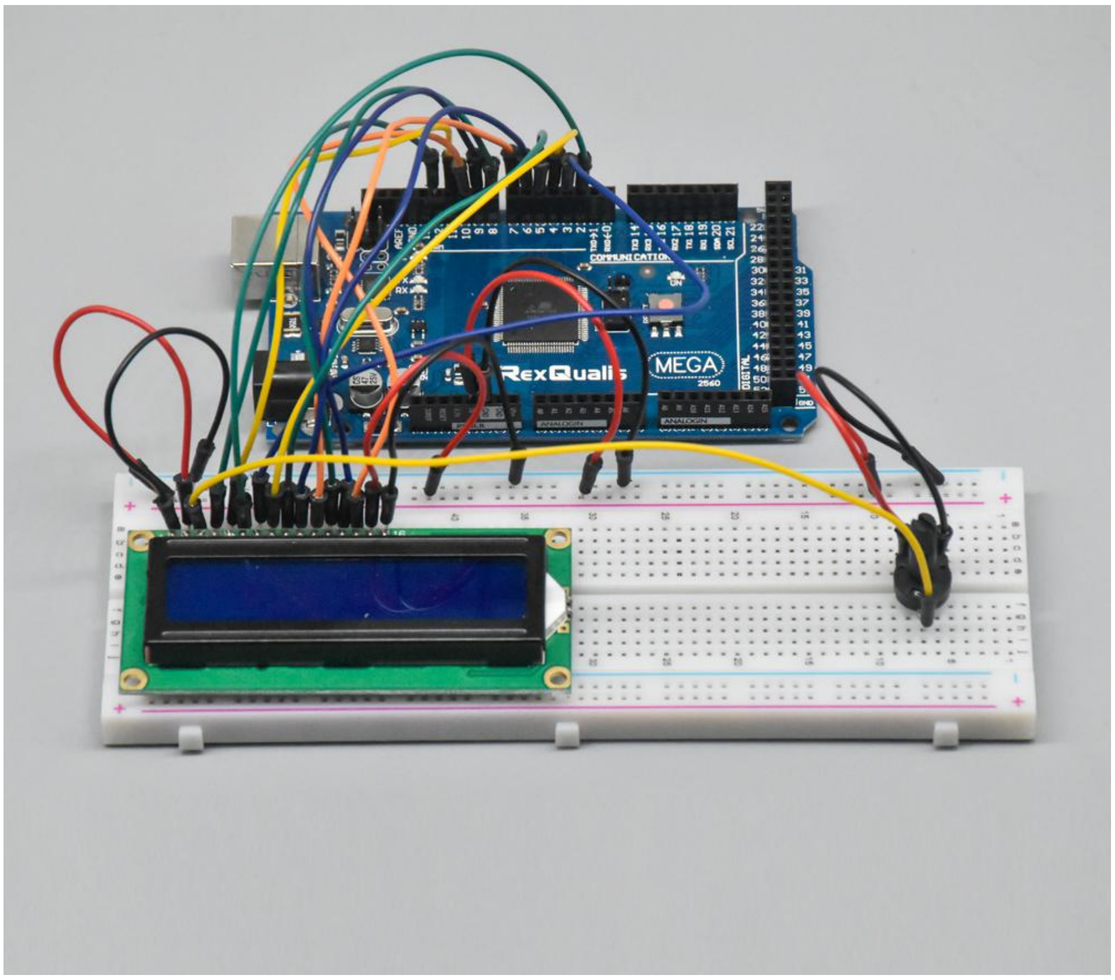
对象类型(T): All Files (*.*)

Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM123

Step 3:Attach Arduino Mega 2560 board to your computer via USB cable and check that the 'Board Type' and 'Serial Port' are set correctly.

Step 4:Upload the code to the RexQualis Mega 2560 board.

Then, you can see on the LCD Display that we have just entered the text "Welcome to, RexQualis……",and you can adjust the brightness of the screen with the Potentiometer.



You can see the video of the experiment results on YouTube:

https://youtu.be/Lm_CJwD8fXE

If it isn’t working, make sure you have assembled the circuit correctly, verified and uploaded the code to your board.For how to upload the code and install the library, check Lesson 0 Preface.