

# CPEN 291 2016W2

## Lab 3

*Lab section: L2A*

*Team Bench #:5B*

<i>Student name</i>	<i>Student number</i>	<i>Contribution percentage</i>
Ziqiao Lin	10668168	33.3%
Yuhao Huang	55562152	33.3%
Yuxiang Huang	14605159	33.3%

### *Contribution summary:*

*Ziqiao Lin:* code for keypad and “Guess the Number” game; part of the lab report.

*Yuhao Huang:* circuit layout; schematic in Fritzing; wiring; part of the lab report;

*Yuxiang Huang:* code for LCD; part of the lab report.

### **A. Introduction and motivations**

The objective of this lab is for us to learn about equipping Arduino with a standalone display using LCD, reading from a keypad, and the coaction between the LCD and keypad. The lab consist of three steps:

1. Connect LCD to Arduino and implement a program to display characters on LCD's screen.
2. Connect the keypad to Arduino and implement a program to read from the keypad.
3. Implement a “Guess the Number” game with the functional LCD and keypad.

### **B. Lab Description**

1. **Display on LCD:** The purpose of using the LCD is to display the information and steps of the “Guess the Number” game. We are expected to initialize and use the LCD without using the liquidCrystal.h library, so we implemented 7 functions to achieve this purpose: LCD\_Write, LCD\_Char, LCD\_String LCD\_Int, LCD\_Clear, LCD\_Initial, and LCD\_SetCursor. At the beginning of the program, a set of initialization commands are sent to the LCD using the function LCD\_Initial. After the initialization, the LCD begins to display the texts (e.g. “Enjoy the game!”) and the numbers. It refreshes itself every time it needs to display new contents, using LCD\_Clear and delay functions. Also, it stops refreshing for new contents when waiting for user inputs. In addition, we implemented an extra feature that the LCD will show the correct number if the player loses the game.

2. **Read from Keypad:** We defined two arrays of the numbers of pins connected to each row and column respectively, with rows being output and columns being input. All output are initially high because they are connected with internal pull-up resistor. We wrote a method called keypad which returns the number that we pressed on the keypad. When any button is pressed, the function determines which row has turned to low. After finding the row number, the function resets the input to high. It also finds the number of the column which causes the row number to change from low to high. The return value is row number \* 3 + column number + 1, which corresponds to the number pressed on the keypad.
3. **Guess the Number game:** This part of the lab is to implement a game which allows user to choose a difficulty level (represented by the range of the number to guess from, level 1 from 0 to 20, level 2 from 0 to 50, and level 3 from 0 to 99) and guess a number. The user wins if the number entered by user equals the random number picked by the program, and lose otherwise. Every text and number will be displayed on the LCD, so this part is actually a coaction between the LCD display and the keyboard. The game first prompts user for a difficulty level, and does not continue until user enters a number between 1 and 3. Then the program generates a random number in the range, and prompts user for a guess. We use a while loop to check the time user has guessed, with the max guess chance being 5. Each time the program goes through the loop it checks whether the guessed number is equal to, less than, or greater than the random number, and prints the result on the LCD; if the numbers are equal the loop terminates and reset everything to start a new game.
4. **Circuit:** We created the circuit as a whole according to the datasheets and instructions in lecture slides. The 7 pins of the keypad is connected to 7 digital pins respectively. The LCD is connected to ground, voltage supply, 4 analog pins, and a potentiometer.

### C. Conclusions and Reflections

Overall, the program succeeded in both displaying on LCD and reading from the keypad. A major problems we encountered during this lab was that we spend lots of time on researching online about the LCD, because we couldn't understand its context easily and quickly. As a result, we nearly ran out of time before the demo time. In case of this happening again in the future, we should start the lab earlier next time, and enhance the communication among group members to increase efficiency.

#### D. References

- Include any cross-references you have used.

Arduino Reference for the functions: <http://arduino.cc/en/Reference/HomePage>

LCD datasheet: [TC1602A-01T.pdf](#)

LCD's extra POT: <http://www.digikey.com/product-detail/en/3386F-1-103TLF/3386F-103TLF-ND/1232544>

LCD commands descriptions: <http://mil.ufl.edu/4744/docs/lcdmanual/commands.html>

Keypad library: <http://playground.arduino.cc/Code/Keypad#Download>

Keypad info: <http://www.digikey.ca/product-detail/en/419/1528-1136-ND/5353596>

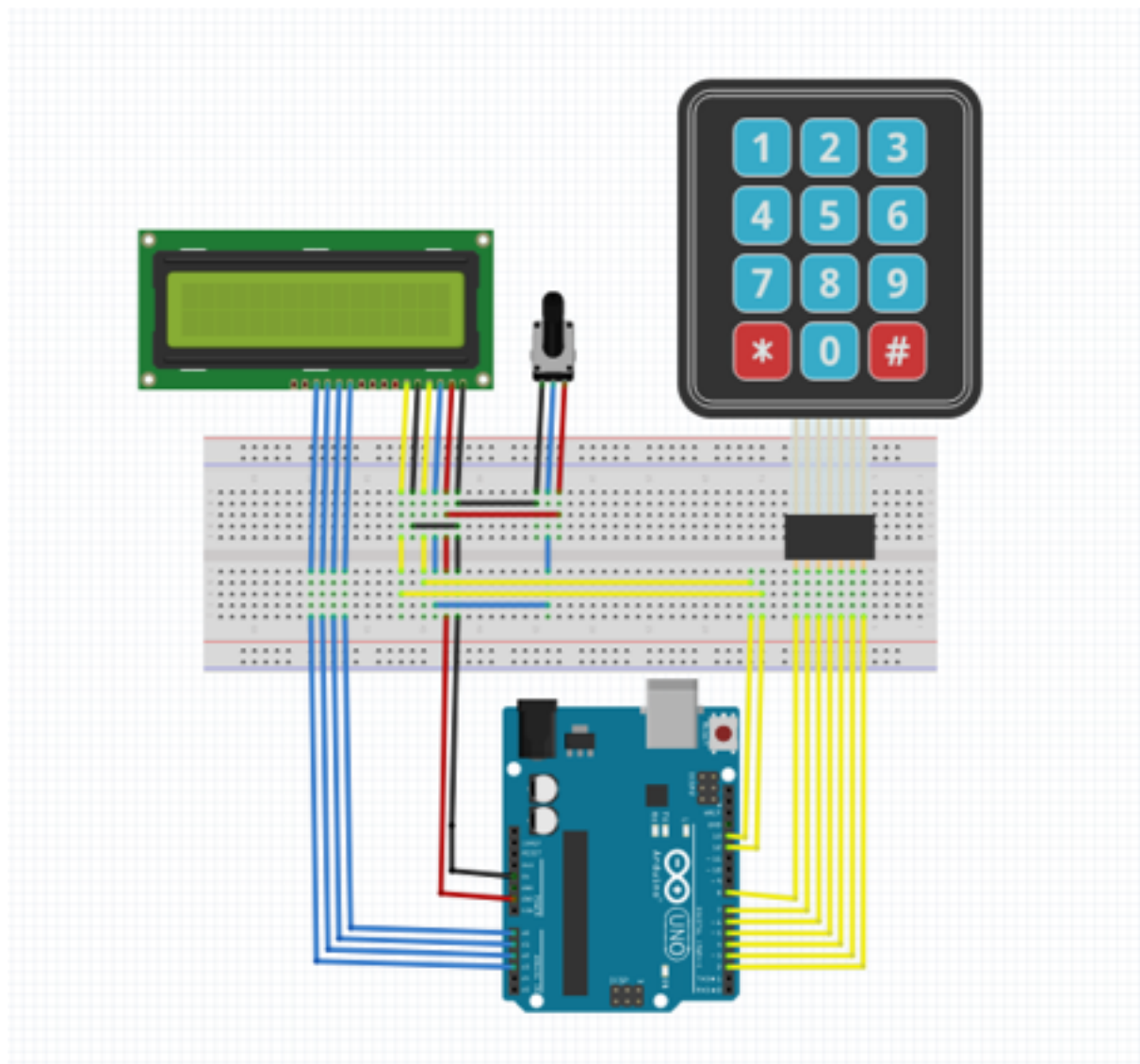
Arduino LCD TutorialHow To Contro lAn LCD <https://www.youtube.com/watch?v=dZZynJLmTn8&t=3s>

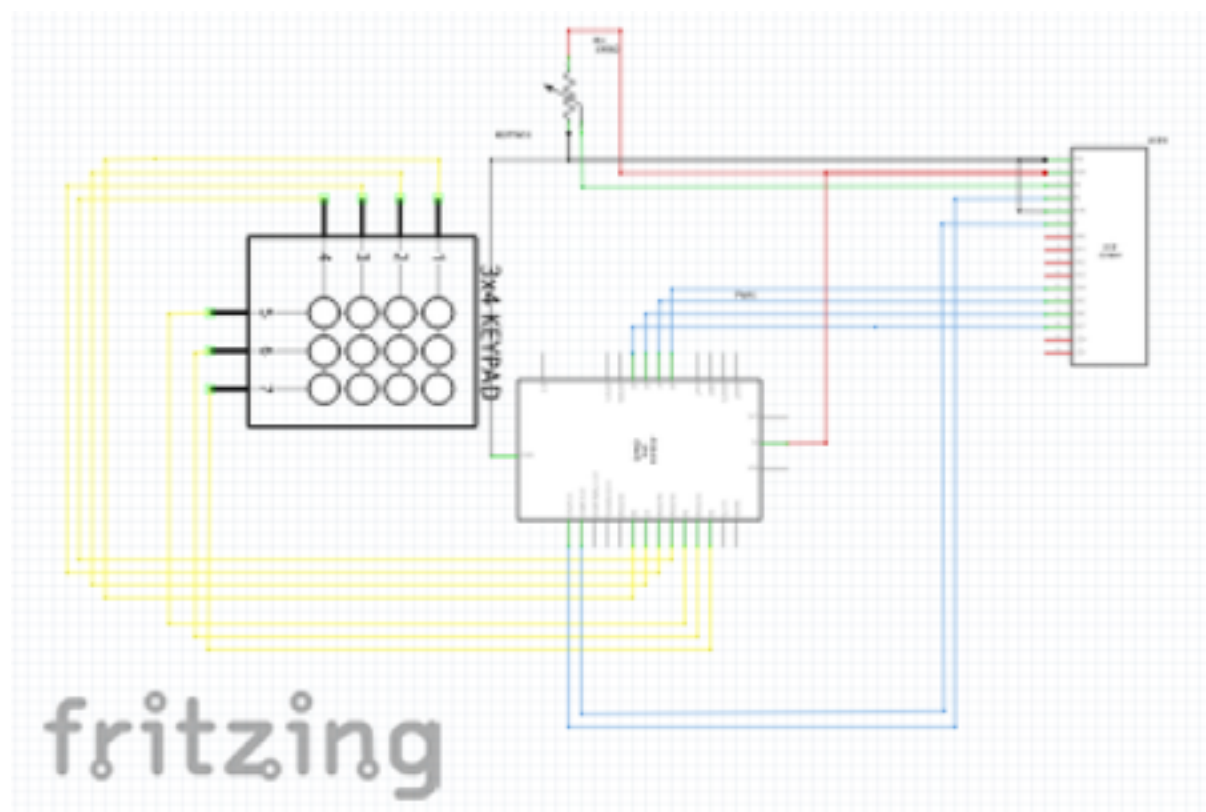
Random Number Geneator: <https://www.youtube.com/watch?v=QCe2ks9b8YI&t=276s>

How to stop void loop: <https://www.youtube.com/watch?v=AuiWwJZQEec&t=711s>

Our LCD is based on this: <http://forum.arduino.cc/index.php?topic=287644.0>

## Appendix I (Schematics on Fritzing)





## Appendix II (codes)

```
/*
 * Author: Yuxiang Huang, Ziqiao Lin, Yuhao Huang
 * Date: January 26, 2017
 * Purpose: Implement the "Guess the Number" Game; learn to
 * use keypad and LCD without library; Finish the lab 3 of
 * CPEN 291
 */

//Pins for keyboard
const int rows = 4; //number of rows of keypad
const int columns = 3; //number of columns of keypad
const int Output[rows] = {8,7,6,5}; //array of pins used as output for rows of keypad
const int Input[columns] = {4,3,2}; //array of pins used as input for columns of keypad

//Pins for LCD
const int D4 = A0;
const int D5 = A1;
const int D6 = A2;
const int D7 = A3;
const int RS = 13;
const int EN = 12;

//Other related parameters
int colNum = 0, rowNum = 0; // two int type to determine the specified column number and
row number
boolean replay = true;
boolean difficultyFlag = 0;

void setup()

{
  for (byte i = 0; i < rows; i++) //for loop used to make pin mode of outputs as output
  {
    pinMode(Output[i], OUTPUT);
  }
  for (byte s = 0; s < columns; s++) //for loop used to make pin mode of inputs as inputpullup
  {
    pinMode(Input[s], INPUT_PULLUP);
  }
  Serial.begin(9600); //to use serial monitor we set the baud rate

  pinMode(RS, OUTPUT);
  pinMode(EN, OUTPUT);
  pinMode(D4, OUTPUT);
  pinMode(D5, OUTPUT);
  pinMode(D6, OUTPUT);
  pinMode(D7, OUTPUT);
}
```

```

    delay(50);
    LCD_Initial();
}

void loop()

{
    //Initialize the cursor position at first
    LCD_SetCursor(0,0);

    //refresh the screen and start/restart a new round of game
    if(replay){
        LCD_Clear();
        LCD_String("enjoy the game !");
        delay(2000);
        guessTheNum();
    }
}

/*****

/*
 * This function is same function as keypad library, which only will determine the number
 * return number = row * 3 + column + 1
 */
int keypad()

{

    bool no_press_flag = 0;                // flag used to ensure no button is pressed
    for (byte x = 0; x < columns; x++)      // for loop used to read all inputs of keypad to
    ensure no button is pressed
    {
        if (digitalRead(Input[x]) == HIGH);    //read evry input if high continue else break;
        else
            break;
        if (x == (columns - 1))                //determine x value to make sure if no button is
    pressed
    {
        no_press_flag = 1;                    // set flag as 1
        colNum = 0;
        rowNum = 0;
    }
    }
    if (no_press_flag == 1)                  //if no button is pressed
    {
        for (byte r = 0; r < rows; r++)        //for loop used to make all output as low
            digitalWrite(Output[r], LOW);
        for (colNum = 0; colNum < columns; colNum++)    // for loop to check if one of inputs is
    low

```

```

{
    if (digitalRead(Input[colNum]) == HIGH)          //if specific input is remain high (no press
on it) continue
        continue;
    else //if one of inputs is low
    {
        for (rowNum = 0; rowNum < rows; rowNum++)    //for loop used to determin the row
number which make switch on
        {
            digitalWrite(Output[rowNum], HIGH);      //make specified output as HIGH then
determine which row number
            if (digitalRead(Input[colNum]) == HIGH)  //makes the column number HIGH
            {
                no_press_flag = 0;                  //reset the flag to low;
                for (byte w = 0; w < rows; w++)      // make all outputs as low
                    digitalWrite(Output[w], LOW);
                return rowNum * 3 + colNum + 1;      //return number of button
            }
        }
    }
}
return 50;
}

```

```

/*****
//The following functions are for the keypad
/*
* This function is the guess the number game; the game will terminatate if the user wins
* or loses, and start a new round afterwards
*/
void guessTheNum(){
    int range = difficulty(); // get difficulty level
    do{
        LCD_Clear();
        delay(200);
        LCD_SetCursor(0, 0);
        LCD_String("checking");
        delay(800);
    }while(!difficultyFlag);
    int attemptNum = 5;      // attempt number
    int counter = 0;        // set counter from 0
    LCD_SetCursor(1, 0);
    LCD_String("take a guess");
    LCD_Clear();
    LCD_SetCursor(0,0);
    LCD_String("take a guess");
    int guessNum = 0;       // define a guess number
    int ranNum = rand()%range; // get ranNum in the range from difficulty level

```



```

bool judge = 0;          // set a flag which will determine when will jump off the loop

do{                      // the loop to get the result from game
    if( counter <= attemptNum){    // if counter <= attemptNum, go inside the condition

        guessNum = getNum();
        if(guessNum == ranNum){    // guess number == random number, the game over, user
won
            LCD_Clear();
            delay(200);
            LCD_SetCursor(0, 0);
            LCD_String("You won!");
            delay(2000);
            judge = 1;
            difficultyFlag = 0;
        }
        if(guessNum > ranNum){    // guess number > random number, notice user to guess
lower
            LCD_String(" Guess Lower");
            counter++;           // counter + 1
        }
        if(guessNum < ranNum){    // guess number < random number, notice user to guess
higher
            LCD_String(" Guess Higher");
            counter++;           // counter + 1
        }
    }else{                // if counter > attempt number, game over, and user lose
        LCD_Clear();
        delay(200);
        LCD_SetCursor(0, 0);
        LCD_String("You lost!");
        delay(2000);

        LCD_Clear();
        delay(200);
        LCD_SetCursor(0, 0);
        LCD_String("The number is: ");
        LCD_SetCursor(1, 0);
        LCD_Int(ranNum);
        delay(2000);

        judge = 1;
        difficultyFlag = 0;
    }
}while(!judge);

return 0;
}
//*****//

```

```

/*
 * getNum() function will return a number which user press on keypad
 */
int getNum(){
    int num = 0; // define a number as num which will be returned to game
    int commandKey = 0; // this number will receive command from keypad
    commandKey = keypad();

    /*
     * this loop will continue to receive input until user press "#"
     */
    while(commandKey !=12){          // case 12 means "#" in keypad, which represent
"enter" case
        switch (commandKey)          //switch used to specify which button
        {
            case 1:                  // case 1 to case9 represent 1 - 9 in keypad
            case 2:
            case 3:
            case 4:
            case 5:
            case 6:
            case 7:
            case 8:
            case 9:
                num = num*10 + commandKey; // when we receive 2 or more number, we need
to represent as format xxxxx
                LCD_Clear();
                LCD_SetCursor(0, 0);
                LCD_String("your guess: ");
                LCD_SetCursor(1, 0);
                LCD_Int(num);
                break;

            case 11:                  // case 11 represent "0" in keypad
                num = num * 10 + 0;    // num + 0 in total number
                LCD_Clear();
                LCD_SetCursor(0, 0);
                LCD_String("your guess: ");
                LCD_SetCursor(1, 0);
                LCD_Int(num);
                break;

            case 10:                  // case 10 represent "*" in keypad which will make one
backspace
                num = num/10;          // use total number / 10 will eliminate the last digit in
num

                LCD_Clear();
                LCD_SetCursor(0, 0);
                LCD_String("your guess: ");
                LCD_SetCursor(1, 0);

```

```

        LCD_Int(num);
        break;
    }
    commandKey = keypad();    // receive next input from keypad
}
LCD_Clear();
delay(500);
LCD_SetCursor(0, 0);
LCD_String("your guess: ");
LCD_SetCursor(1, 0);
LCD_Int(num);
return num;
}

/*
 * this function is for user to choose different difficulties for the guess number game
 * return int option, which represent the number that will be rand() rest.
 */
int difficulty(){

    //Clear the LCD for a new display
    LCD_Clear();
    LCD_SetCursor(0, 0);
    LCD_String("difficulty Lv: "); // print difficulty for user
    delay(2000);

    LCD_Clear();
    delay(200);
    LCD_SetCursor(0, 0);
    LCD_String("1: range 0 - 20"); // difficulty level 1 for 0 - 20
    delay(2000);

    LCD_Clear();
    delay(200);
    LCD_SetCursor(0, 0);
    LCD_String("2: range 0 - 50"); // difficulty level 2 for 0 - 50
    delay(2000);

    LCD_Clear();
    delay(200);
    LCD_SetCursor(0, 0);
    LCD_String("3: range 0 - 99"); // difficulty level 3 for 0 - 99
    delay(2000);

    //Clear the LCD screen for the display later
    LCD_Clear();
    int option = 0; // define a number as int
    int command = 0;

```

```

//Get the input from the keypad, and show the difficulty level on the LCD screen
do{
  command = keypad();
}while(command<1 || command > 3);
switch(command){
  case 1: option = 21;
    LCD_String("Lv 1");
    delay(1000);
    break;    //which can create random number 0 - 20
  case 2: option = 51;
    LCD_String("Lv 2");
    delay(1000);
    break;    //which can create random number 0 - 50
  case 3: option = 100;
    LCD_String("Lv 3");
    delay(1000);
    break;    //which can create random number 0 - 99
}
difficultyFlag = 1;
return option; // return number
}

/*****
*****Funntions For LCD*****
*****/

/*This functino is used to write commands/data to the LCD*/
void LCD_Write(byte cmd,int data_cmd){

  //At first, set the RS pin to corresponding value in order to receive instructions or data from
  the Arduino
  //Set RS pin to 1 to receive data, or 0 to receive instructions
  digitalWrite(RS,data_cmd);

  //Since we are using 4-bit mode, we are using buses 4 to 7, and
  //we need to transmit the 8-bit command/data in two steps
  //First, we send the 4 most significant bits to bus 4 to 7 as following*/
  digitalWrite(D4,(cmd & 0b00010000)>>4);
  digitalWrite(D5,(cmd & 0b00100000)>>5);
  digitalWrite(D6,(cmd & 0b01000000)>>6);
  digitalWrite(D7,(cmd & 0b10000000)>>7);

  //delay 1 ms to wait for the completion of data/instruction tranmission
  delay(1);

  //Set the Enable pin to HIGH, and then LOW, to indicate the start of a new transmission
  digitalWrite(EN,HIGH);
  delay(1);
  digitalWrite(EN,LOW);
  delay(1);

```

```

//Start a new set of transmission just as that above
digitalWrite(D4,(cmd & 0b00000001)>>0);
digitalWrite(D5,(cmd & 0b00000010)>>1);
digitalWrite(D6,(cmd & 0b00000100)>>2);
digitalWrite(D7,(cmd & 0b00001000)>>3);
delay(1);
digitalWrite(EN,HIGH);
delay(1);
digitalWrite(EN,LOW);
delay(1);
}

/*This function is used to print a char on the LCD screen;
  It is mainly used in other functions*/
void LCD_Char(char character){
  LCD_Write(character,1);
}

/*This function is used to print a string on the LCD screen*/
void LCD_String(String str){
  char *cstr = new char[str.length()];
  strcpy(cstr, str.c_str());
  while(*cstr) LCD_Char (*cstr++);
}

/*This functions is used to print a number on the LCD scree*/
void LCD_Int(unsigned long n){
  char c[10];
  sprintf(c, "%d", n);
  for (int i=0;c[i] > 0 ;i++) LCD_Char(c[i]);
}

/*This function is used to clear the LCD screen*/
void LCD_Clear(){
  LCD_Write(0x01,0);
}

/*This function is used to send initialization commands to the LCD*/
void LCD_Initial(){
  LCD_Write(0x33,0);
  //Delay 16 ms after sending every instruction to ensure all the instructions
  //have been executed properly
  delay(16);
  LCD_Write(0x32,0);
  delay(16);
  LCD_Write(0x28,0);
  delay(16);
  LCD_Write(0x0C,0);
  delay(16);
}

```

```
LCD_Write(0x06,0);  
delay(16);  
LCD_Write(0x01,0);  
delay(16);  
}
```

```
/*This function is used to set the position of the cursor on the LCD*/  
void LCD_SetCursor(int col, int row){  
    if (col == 0)LCD_Write((row | 0x80),0);  
    if (col == 1)LCD_Write((row | 0x80)+ 0x40,0);  
    if (col == 2)LCD_Write((row | 0x80)+ 0x10,0);  
    if (col == 3)LCD_Write((row | 0x80)+ 0x50,0);  
    delay(1);} 
```