Dumptruck’s Mechatronics Lab Report 2

01.28.22

Nick Morse

Ryan Colon

Summary:

Lab 2 required us to print a recipe to an LCD screen and incorporate a timer into one of the steps, which is displayed via a group of 4 7 segment displays. For this lab we maintained our code from the first lab and added a few lines to incorporate the functionality of the 7-segment timer display and timer. This was done without the use of any extra libraries but did require us to start using the gpio pins.

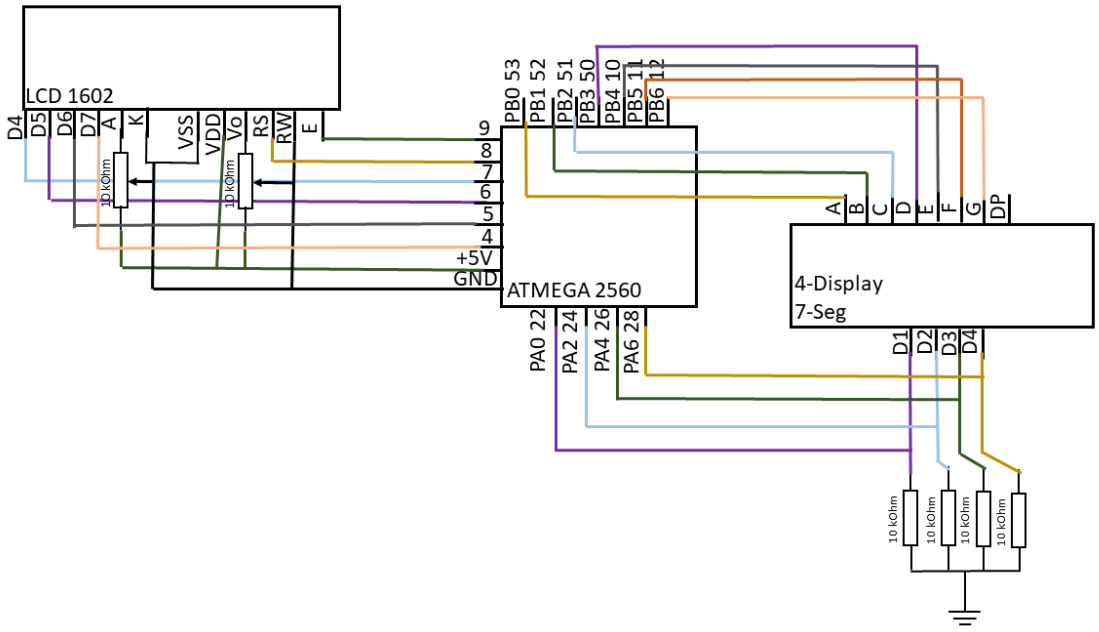
When creating the circuit for this lab, we mostly left the LCD untouched, but we did add a potentiometer to pin V0 of the LCD so we could manually adjust the contrast. We then hooked up the 4-digit 7 segment part, which to control the digit select we used PORTA, and to control the data lines we used PORTB. The digit select pins (D1-D4) were connected to ground via pull down resistors to ensure a complete for each of the 7 segments.

It's important to note that via this implementation, it is only possible to finely control a single digit of the 7-segment group at a time, so they must be switched between quickly to create the illusion that they are displaying different numbers on all screens at once. The time it takes to display to all the digits needs to be less than 20 ms total before blinking is noticeable to the human eye, but it cannot be instant because all the digits will then blur together and be impossible to read. In our code this manifests as a 1ms delay between writing to each 7-segment, which is sufficiently fast enough for all the digits to look distinct and for no blinking to appear. An oscilloscope was taken to the D4 digit enable line to measure the waveform:

Figure 1. Oscilloscope Output

What the output of the oscilloscope tells us is that each digit blinks roughly at 250 Hz, which is too fast for the human eye to make out. And indeed, when tested the blinking was imperceptible. To implement the timer: we used the millis() Arduino function to keep track of time, which then decremented a ‘seconds’ integer every time a second passed, this ‘seconds’ integer was then converted to (MM:SS) format and then displayed to the 7-segments. After the timer portion had ceased, the program continued and completed as it had the first lab.

Circuit Diagram:



Code:

```
/*
Intelligent Machines Lab 1 LCD Recipe
Ryan Colon
01.16.22
 */

#include <LiquidCrystal.h>

const int rs = 8, en = 9, LCDd4 = 7, LCDd5 = 6, LCDd6 = 5, LCDd7 = 4;
LiquidCrystal lcd(rs, en, LCDd4, LCDd5, LCDd6, LCDd7);

const int upperCPMBound = 800, lowerCPMBound = 700;

int upperCPms, lowerCPms;

String currLowerMessage;

/* 7 Segments stuff
 *   PA0    PA2    PA4    PA6
 *  D1 = 22, D2 = 24, D3 = 26, D4 = 28
 *   PB0   PB1   PB2   PB3   PB4   PB5   PB6
 *  a = 53, b = 52, c = 51, d = 50, e = 10, f = 11, g = 12
 */

//Address for each 7-segment of the 47S display
const int SSD1 = 0b1111110, SSD2 = 0b1111011, SSD3 = 0b1101111, SSD4 = 0b0111111; //The devices are active low, no I don't feel like fixing that
```

```
//OPCodes for displaying integer N in SSN on 7-segment display

const int SS9 = 0b1101111, SS8 = 0b1111111, SS7 = 0b0000111, SS6 = 0b1111101, SS5 = 0b1101101, SS4 = 0b1100110, SS3 = 0b1001111, SS2 = 0b1011011, SS1 = 0b0000110, SS0 = 0b0111111, SSN = 0b0000000;

const int rotateDelay = 1; //delay in ms between changing 7 segments

const int SSC[11] = {SS0, SS1, SS2, SS3, SS4, SS5, SS6, SS7, SS8, SS9, SSN};


void setup() {
  DDRB = 0b11111111; // set all PORTB as outputs
  DDRA = 0b11111111; // set PA0, PA2, PA4, PA6 as outputs


  lcd.begin(16,2);
  lcd.clear();
  Serial.begin(9600);


  upperCPms = ceil((60.0 * 1000.0) / (float) upperCPMBound);
  lowerCPms = ceil((60.0 * 1000.0) / (float) lowerCPMBound);


  Serial.print("upper bound:\t");
  Serial.println(upperCPms);
  Serial.print("lower  bound:\t");
  Serial.println(lowerCPms);


  typeToLCD("DMPTRKS Recipe:");
  lcd.setCursor(0,1);
  typeToLCD("Cinnamon Toast!");
  delay(2500);
  lcd.clear();
  typeToLCD("You'll need:");
```

```
lcd.setCursor(0,1);
typeToLCD("  A toaster");
currLowerMessage = "  A toaster";
delay(1500);
vertScrollLCD("  1 slice bread");
delay(1500);
vertScrollLCD("  Some butter");
delay(1500);
vertScrollLCD("  Some cinnamon");
delay(2500);
lcd.clear();
lcd.setCursor(0,0);
lcd.print("   Let's get    ");
lcd.setCursor(0,1);
lcd.print("   toasting!    ");
delay(2000);
LCDStepFormatting(1, "  get bread");
delay(2500);
LCDStepFormatting(2, "bread in toaster");
delay(2500);
LCDStepFormatting(3, "  toast bread");
delay(2500);
LCDStepFormatting(4, "  wait for toast");
timerLoop(90);
PORTB = SSC[10];
delay(1000);
LCDStepFormatting(5, "  butter time");
delay(2500);
LCDStepFormatting(6, "butter the toast");
```

```
  delay(2500);
  LCDStepFormatting(7, "  spray cinnamon");
  delay(2500);
  LCDStepFormatting(8, "  feed \"mother\"");
}

void loop() {

}

//Fumction to display to 7-seg a timer and countdown, takes number of seconds as an integer input
void timerLoop(int numSec){

  unsigned long int currTime = millis();
  bool startBool = true;
  int nums[4];

  while(numSec != 0){
    if(millis() - currTime >= 1000 || startBool){
      startBool = false;
      int* digits = intToTime(numSec);

      for(int i = 0; i < 4; i++)
        nums[i] = digits[i];

      delete [] digits;
      --numSec;
      currTime = millis();
    }
```

```
    PORTA = SSD1;
    PORTB = SSC[nums[0]];
    delay(rotateDelay);
    PORTA = SSD2;
    PORTB = SSC[nums[1]];
    delay(rotateDelay);
    PORTA = SSD3;
    PORTB = SSC[nums[2]];
    delay(rotateDelay);
    PORTA = SSD4;
    PORTB = SSC[nums[3]];
    delay(rotateDelay);
  }
  delay(1000);
}

//Function to convert integer to time digits, returns an integer array of size 4
int* intToTime(int seconds){
  int* digits = new int[4];

  digits[1] = floor(seconds/60);
  digits[0] = floor(digits[1]/10);
  digits[1] = digits[1]%10;

  digits[3] = seconds%60;
  digits[2] = floor(digits[3]/10);
  digits[3] = digits[3]%10;

  return digits;
```

```
}

/*
Function to display text to lcd using a vertical scrolling effect
You need to initialize currLowerMessage before using
If you see the lower text before scrolling not mention the top text after scrolling
then you need to double check and make sure currLowerMessage is being used properly
*/
void vertScrollLCD(String textToPrint){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(currLowerMessage);
  lcd.setCursor(0,1);
  typeToLCD(textToPrint);
  currLowerMessage = textToPrint;
}

//Function that takes a string and writes it to the LCD in a typing style
void typeToLCD(String toType){
  for(int i = 0; i < toType.length(); i++){
    lcd.print(toType[i]);
    delay(random(lowerCPms, upperCPms));
  }
}

//Function to print string in text formatting
//can not accept a stepNum value > 99
void LCDStepFormatting(int stepNum, String toPrint){
  String stepString = "Step ";
```

```
  char temp[2];
  itoa(stepNum, temp, 10);
  stepString += temp;
  stepString += ':';

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print(stepString);
  lcd.setCursor(0,1);
  typeToLCD(toPrint);
}
```