

Objective: Setting up the library software and using it to control the LCD.

Prelab: The code was written and uploaded to the website. The code written and seen below is a simple program that counts and displays the number of seconds since the program has started on the LCD.

Part 1) In part one, the code was uploaded to the Arduino and the timer was displayed on the LCD.

Part 2) In part 2, the program was altered to, in addition to the timer in part one, have a second timer that is activated when the encoder knob is pushed. This timer would count in 100 millisecond time intervals and counts as long as the knob is pressed. Using `LcdDriver.setCursor()` the first timer is displayed on line one and the 100 millisecond timer on the second line. When released the LCD displays the length of time the button was pushed. The original timer continues as long as the program is running. The code for part two can be seen below.

Questions:

1) How could you test your stop watch like timer?

By comparing it to a more reliable stopwatch such as high end mechanical or accurate digital watch.

2) Would want to let the stop watch for a long or short time period, and WHY?

A short time, because this is not the most accurate simulations of a digital stopwatch the timer is not 100% accurate. Over a longer period of time those small inaccuracies would compound and throw the accuracy of the stop watch off. Over a short period of time those inaccuracies would be minute.

Appendix A) Code for Part 1

```
// Import LCD Library
#include <LiquidCrystal.h> // Must be included in you code.

// Defines for LCD
LiquidCrystal LcdDriver(11, 9, 5, 6, 7, 8 ); // Must be in your code, exactly
like this.
int count; //used to count the number of seconds that have passed
unsigned long LedTimer; //unsigned long to keep track of system time in
milliseconds

void setup() {
    LcdDriver.begin(16,2); // begin
    LcdDriver.clear();// clear
    count = 0; //sets count = 0
    LedTimer = millis(); //Sets the LedTimer to the amount of milliseconds
since the program started
}

void loop() {

    if ( millis() - LedTimer >= 1000) { //checks if at least one second has
passed since the last loop
```

```
        count += 1;           //increments the count by 1
        LcdDriver.clear( );
        LcdDriver.print(count); //prints the new count to the LCD
        LedTimer += 1000; //increments the LedTimer by 1000 milliseconds
    }
}

Appendix B for part 2)

// Import LCD Library
#include <LiquidCrystal.h> // Must be included in you code.

// Defines for LCD
LiquidCrystal LcdDriver(11, 9, 5, 6, 7, 8 ); // Must be in your code, exactly
like this.
int count; //used to count the number of seconds that have passed
unsigned long LedTimer; //unsigned long to keep track of system time in
milliseconds
int count2; //used to count the number of seconds that have passed
unsigned long Stopwatch;

void setup() {
    LcdDriver.begin(16,2); // begin
    LcdDriver.clear();// clear
    count = 0; //sets count = 0
    count2 = 0;
    LedTimer = millis(); //Sets the LedTimer to the amount of milliseconds
since the program started
    Stopwatch = millis();
}

void loop() {
    if ( millis() - LedTimer >= 1000) { //checks if at least one second has
passed since the last
        count += 1;           //increments the count by 1

        LcdDriver.setCursor(0,0);
        LcdDriver.print(count); //prints the new count to the LCD
        LedTimer += 1000; //increments the LedTimer by 1000 milliseconds
    }
    if (millis() - Stopwatch >= 100) {
        if (digitalRead(4) == HIGH) {
            count2 = 0;
        }
        else {
            count2 += 1;
            LcdDriver.clear();
            LcdDriver.print(count);
            LcdDriver.setCursor(0,1);
            LcdDriver.print(count2);
        }
        Stopwatch +=100;
    }
}
```