


Lab 4 - Using the LCD Display

Lab 12:30 W 2:00 W 3:30 W
 Section 11:00 F 12:30 F

TA/Instructor initials: _____


Item	Outcome	Score	Max.
Checkpoint <ul style="list-style-type: none"> • Showed correct LCD enable (“E”) signal on oscilloscope 	Yes/No		1
LCD and Button Operation <ul style="list-style-type: none"> • UP and DOWN buttons change count direction correctly (doesn’t reset to 0) • Count sequence is correct • Count rolls over at end of sequence • PAUSE button works correctly • Counter responds to short button presses 	Yes/No		2
	Yes/No		1
	Yes/No		1
	Yes/No		1
	Yes/No		2
Review Questions (graded after submission) <ul style="list-style-type: none"> • Questions below (put answers in Lab4_Answers.txt file and submit on Vocareum) 			3
Code Organization (Graded after submission) <ul style="list-style-type: none"> • Code is indented properly and includes comments • DDR and PORT bits initialized • LCD initialized • Code to check buttons is correct • State machine used for count direction • Count values changed correctly • lcd_writenibble only changes PORTD[7:4] and PORTB[1:0] 			2
			1
			1
			1
			1
			2
			2
Total			21
Open ended comments:			

Review Problems

- (2 points) Normally we only read the bits in the PIN register, but a close reading of the ATmega328P datasheet reveals that **writing a 1 to PIN register bit** has the effect of inverting the bit in the corresponding PORT register. For example, if you write a 1 to PINC, bit 3, it will invert bit 3 in the PORTC register. Based on this information, Billy Bruin has decided that he now knows an easy way to “toggle” the E bit (flipping it to its opposite value) to generate the E pulse in the lcd_writenibble() function by using this code.

```
PINB |= (1 << PB1)    // Toggle E bit from 0 to 1
PORTB |= (1 << PB1);  // Delay to make the E pulse longer
PINB |= (1 << PB1)    // Toggle E bit from 1 to 0
```

Note: `PINB |= (1 << PB1);` is equivalent to `PINB = PINB | (1 << PB1);`

Tammy Trojan has also read the datasheet and found that when reading the PIN register, if a bit in the group is configured as input, then the voltage coming in the PIN is returned, but if a bit is configured as output, **the corresponding PORT bit value is returned**. From this she concludes that it is possible that Billy's method can cause problems depending on how the compiler converts the program to instructions the processor executes and therefore should not be used. For the program to work reliably, she recommends using this code to generate the E pulse.

```
PINB = (1 << PB1)      // Toggle E bit from 0 to 1
PORTB |= (1 << PB1);   // Delay to make the E pulse longer
PINB = (1 << PB1)      // Toggle E bit from 1 to 0
```

Tammy says that in this lab assignment with the PAUSE button on PB3, Billy Bruin's code can cause the PAUSE button to stop working. Can you explain why this could happen?

2. (1 point) Suppose we need to perform 3 concurrent tasks intermittently: Task A every 20 ms, Task B every 15 ms, and Task C every 40 ms. What delay should be used on each iteration of the main loop?