

ECE3411 – Fall 2015

Lab 3c.

# Non-Blocking LCD (Extended State Machine)

---

**Marten van Dijk, Syed Kamran Haider**  
Department of Electrical & Computer Engineering  
University of Connecticut  
Email: {vandijk, syed.haider}@engr.uconn.edu

**UConn**



# LCD Blocking Data Write Example

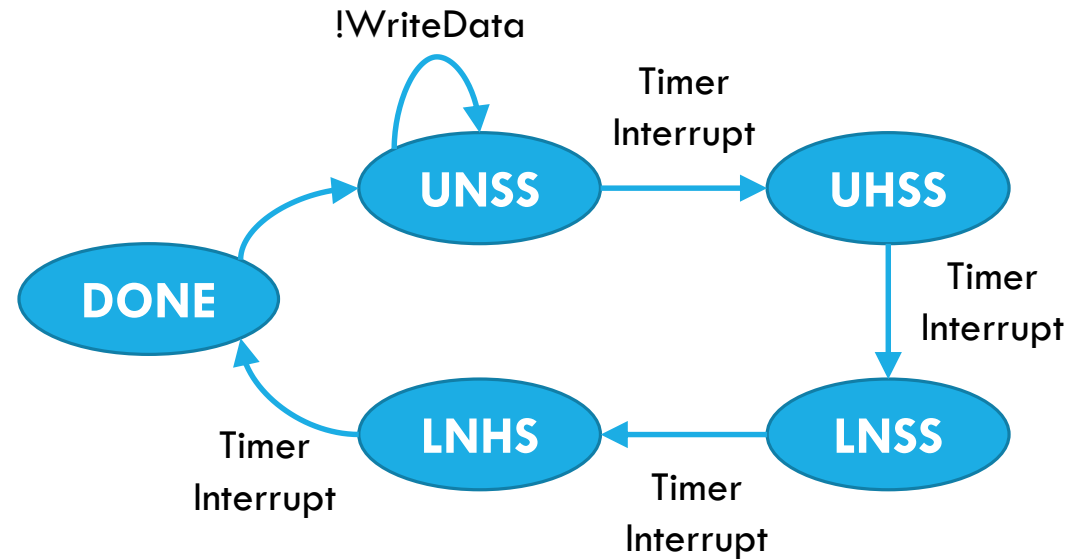
```
void LcdDataWrite(uint8_t da)
{
    // First send higher 4-bits
    DATA_PORT = (DATA_PORT & 0xf0) | (da >> 4);
    CTRL_PORT |= (1<<RS);
    CTRL_PORT |= (1<<ENABLE);
    _delay_ms(1); // WASTED CYCLES
    CTRL_PORT &= ~(1<<ENABLE);
    _delay_ms(1); // WASTED CYCLES

    // Send lower 4-bits
    DATA_PORT = (DATA_PORT & 0xf0) | (da & 0x0f);
    CTRL_PORT |= (1<<RS);
    CTRL_PORT |= (1<<ENABLE);
    _delay_ms(1); // WASTED CYCLES
    CTRL_PORT &= ~(1<<ENABLE);
    _delay_ms(1); // WASTED CYCLES
}
```

# LCD Non-Blocking Write

In last lab, we implemented the following state machine:

- Split the Blocking-write into 4 states
  - Upper Nibble Setup State (UNSS)
  - Upper Nibble Hold State (UHSS)
  - Lower Nibble Setup State (LNSS)
  - Lower Nibble Hold State (LNHS)
- Transition to next state upon timer interrupt
  - Meanwhile do something else.



# Task 1: Non-Blocking LCD Command+Data Write

---

Implement Non-Blocking LCD Writes using the Timer Interrupts and demonstrate LCD refresh rate of **exactly** 1Hz.

In particular, implement the following:

- Implement a non-blocking *LCD\_GoTo\_and\_Write(int x, int y, char data)* function that writes a LCD GoTo command and a data to the resulting location back to back in a truly non-blocking fashion.
  - Implement a large state machine that can handle two back to back LCD write operations.
- Print a different character on LCD after exactly 1 second to show a refresh rate of 1Hz, e.g. first print '0' then '1' after a second, and so on.
- You are **NOT ALLOWED** to use `_delay_ms()` or `_delay_us()` functions.

# Extended LCD Write State Machine

