

ECE3411 – Fall 2016

Lab 3b.

# Non-Blocking LCD (using Timer ISR)

---

**Marten van Dijk, Chenglu Jin**

Department of Electrical & Computer Engineering

University of Connecticut

Email: {marten.van\_dijk, chenglu.jin}@uconn.edu

Copied from Lab 3b, ECE3411 – Fall 2015, by  
Marten van Dijk and Syed Kamran Haider

**UConn**



# LCD functions so far!

---

- So far the LCD functions are blocking...
  - These functions use `_delay_ms()` or `_delay_us()` routines.
- We need to make use of the wasted CPU cycles
  - Hence we need to get rid of `_delay_ms()` & `_delay_us()` routines
- We use timer interrupts to trigger an event once certain time period has elapsed
  - Meanwhile do other work instead of waiting idle for the time to pass.

# LCD Blocking Command Write Example

```
void LcdCommandWrite(uint8_t cm)
{
    // First send higher 4-bits
    DATA_PORT = (DATA_PORT & 0xf0) | (cm >> 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) | (cm & 0x0f);
    CTRL_PORT &= ~(1<<RS);
    CTRL_PORT |= (1<<ENABLE);
    _delay_ms(1); // WASTED CYCLES
    CTRL_PORT &= ~(1<<ENABLE);
    _delay_ms(1); // WASTED CYCLES
}
```

# 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
}
```

# Task 1: Debounce State Machine with Timer ISR

---

- Implement the LED Frequency Toggling task from Lab3a:Task1 using the Timer Interrupts, i.e.
  - Use the Extended Debounce State Machine to read a Push Switch.
  - On a button push, toggle the LED blinking rate between 2Hz & 8Hz.
  - You don't need to print anything on LCD or UART.
  - You are **NOT ALLOWED** to use `_delay_ms()` or `_delay_us()` functions.

# Task 2: Non-Blocking LCD Writes

---

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

- In particular, implement the following:
  - Non-Blocking LcdDataWrite(uint8\_t data) and LcdCommandWrite(uint8\_t command) functions.
  - Print a different character on LCD after exactly 1 second to show a refresh rate of 1 Hz, 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.