

NMK32203 - Microcontroller

Note20241017 This page is being updated - work in progress...

This course is *Microcontroller*, offered by the Faculty of Electronics Engineering & Technology.

Announcements

[20241017] Updating this for 202425s1 Academic Session.

Lecture Slides

- Lecture 1 - [Overview](#)
- Lecture 2 - [Architecture](#)

Codes (discussed in class)

Introduction to using 'library' code. Single header include file(s) meant for single source compilation. These are meant to be compiled using Keil 8051 compiler.

Library: Serial (UART)

[uart.h](#)

```
#ifndef __MY1UART_H__
#define __MY1UART_H__

/* note: this is NOT compatible with keil's printf */
void uart_init(void) {
    SCON = 0x50;
    TMOD &= 0x0F;
    TMOD |= 0x20;
    TH1 = 253;
    TR1 = 1;
}

void uart_send(unsigned char sdat) {
    SBUF = sdat;
    while (TI==0);
    TI = 0;
}
```

```
unsigned char uart_read(void) {
    unsigned char rdat;
    while (RI==0);
    rdat = SBUF;
    RI = 0;
    return rdat;
}

void uart_puts(char* text) {
    while (*text) {
        uart_send(*text);
        text++;
    }
}

#endif /* __MY1UART_H__ */
```

Library: Text LCD

[tlcd.h](#)

```
#ifndef __MY1TLCD_H__
#define __MY1TLCD_H__

#define LCD_DATA P2
sbit LCD_RS = P0^7;
sbit LCD_RW = P0^6;
sbit LCD_EN = P0^5;

void tlcd_delay(unsigned char step) {
    unsigned int loop;
    do {
        loop = 1000; while (--loop);
    } while (--step);
}

void tlcd_write(unsigned char cdat) {
    LCD_RS = 1;
    LCD_RW = 0;
    LCD_DATA = cdat;
    LCD_EN = 1;
    LCD_EN = 0;
    tlcd_delay(1);
}

void tlcd_cmd(unsigned char ccmd) {
    LCD_RS = 0;
    LCD_RW = 0;
```

```
LCD_DATA = ccmd;
LCD_EN = 1;
LCD_EN = 0;
tlcd_delay(2);
}

void tlcd_init(void) {
    tlcd_cmd(0x38); /* 8 bit mode, 1/16 duty, 5x8 font */
    tlcd_cmd(0x38);
    tlcd_cmd(0x38);
    tlcd_cmd(0x06); /* display off */
    tlcd_cmd(0x0c); /* display on, blink cursor on */
    tlcd_cmd(0x01); /* clear display */
}

void tlcd_puts(char* text) {
    while (*text) {
        tlcd_write(*text);
        text++;
    }
}

#endif /* __MY1TLCD_H__ */
```

Library: Timer

timer.h

```
#ifndef __MY1TIMER_H__
#define __MY1TIMER_H__

// useful macro!
#define timer0_init() { TMOD &= 0xF0; TMOD |= 0x01; }
#define timer0_prep(hi,lo) { TH0 = hi; TL0 = lo; }
#define timer0_stop() { TR0 = 0; TF0 = 0; }
#define timer0_exec() { TR0 = 1; }
#define timer0_flag() { timer0_exec(); while (!TF0); }
#define timer0_wait(hi,lo) { timer0_prep(hi,lo); timer0_flag(); timer0_stop(); }
#define timer0_null() { TH0 = 0; TL0 = 0; }
#define timer0_read() ((unsigned int)TH0<<8)|TL0

void timer0_delays(unsigned int step) {
    do {
        timer0_wait(0xfc,0x66); // 1 ms delay
    } while (--step);
}
```

```
#endif /* __MY1TIMER_H__ */
```

Test Codes

These are not meant for class - just something I use to test boards/modules.

Test Code for Base Kit

[basetest.c](#)

```
/*-----*/
-----*/
/* override default tick values */
#define TIMER_TICK_LEN TIMER_VAL50MS
#define TIMER_TICK_CNT TIMER_LOOP_1S
/*-----*/
-----*/
#include "my1stc51.h"
#include "timer_tick.h"
#include "led7seg.h"
/*-----*/
-----*/
void main(void) {
    unsigned char loop, mask, seg7[] = { _7SEGNUM_CC_ };
    P1MODE1(); // better pullup current
    loop = 0; mask = (!P2_7)?0xff:0x00;
    P1 = seg7[loop] ^ mask;
    timer_init();
    timer_tick_exec();
    while (1) {
        P2_0 = !P3_7;
        P2_1 = !P3_6;
        P2_2 = !P3_5;
        P2_3 = !P3_4;
        if (timer_ticked()) {
            loop++; if (loop==10) loop = 0;
            mask = (!P2_7)?0xff:0x00;
            P1 = seg7[loop] ^ mask;
            timer_tick00();
        }
    }
}
/*-----*/
-----*/
```

From:

<http://azman.unimap.edu.my/dokuwiki/> - **Azman @UniMAP**

Permanent link:

<http://azman.unimap.edu.my/dokuwiki/doku.php?id=archive:nmk322&rev=1729141120>

Last update: **2024/10/17 12:58**

