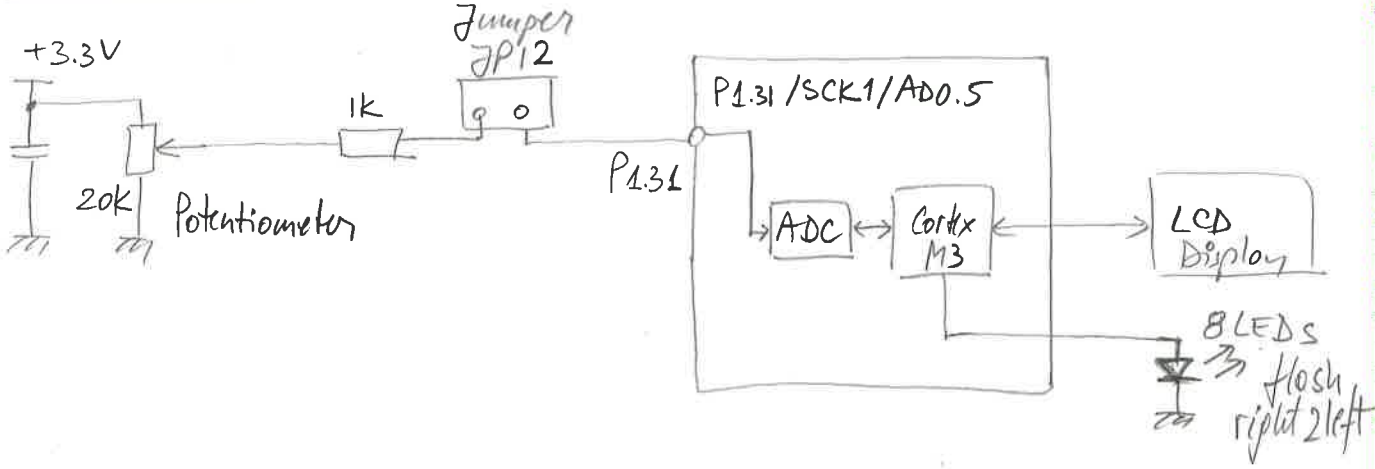
Details on lab #3



SysTick - System Timer

- 24-bit system timer, counts down from reload value to zero, then reloads (wraps to) the value in the LOAD register.
- it runs on the processor clock; times intervals of 10 milliseconds if CPU clock is 100 MHz
- See pages 513, 792 of user manual.
- Basic configuration:
  1. Clock source: either CCLK (internal) or STCLK (external)
  2. Pins
  3. interrupt: system tick Timer interrupt is enabled in NVIC

ADC - Analog to digital Converter (see page 513 of user manual)

- 12 bit successive approximation ADC
- conversion rate 200 kHz
- Basic configuration
  1. power up: PCONP (table 46) `LPC_SC->PCONP |= 1<<12;`
  2. Clock : PCLKSEL ϕ (table 40)
  3. Pins { — Enable ADC ϕ pins thru PINSEL registers
     — Select pin modes thru PINMODE registers
     `LPC_PINCON->PINSEL 3 |= 3<<30;`
  4. interrupts: enable ADC as source of interrupts in NVIC
     `NVIC_Enable IRQ (ADC_IRQn);`
  5. DMA

```
int main (void) {
    ADC_init();
    SysTick_Config (SystemCoreClock/100); //generate interrupt each 10 ms.
    LED_init();
    SER_init (0, 9600);
    while (1) {
        if(AD_done) {
            AD_done = 0;
            ad_avg += AD_last << 8;
            ad_val = (ad_avg >>8) >> 4; //average devided by 16
        }
        if (ad_val ≠ ad_val_prev) {
            LCD_putText( );
            LCD_DrowBargraph( );
        }
        if(clock_1s) {
            SER_putString();
        }
    }
}
```

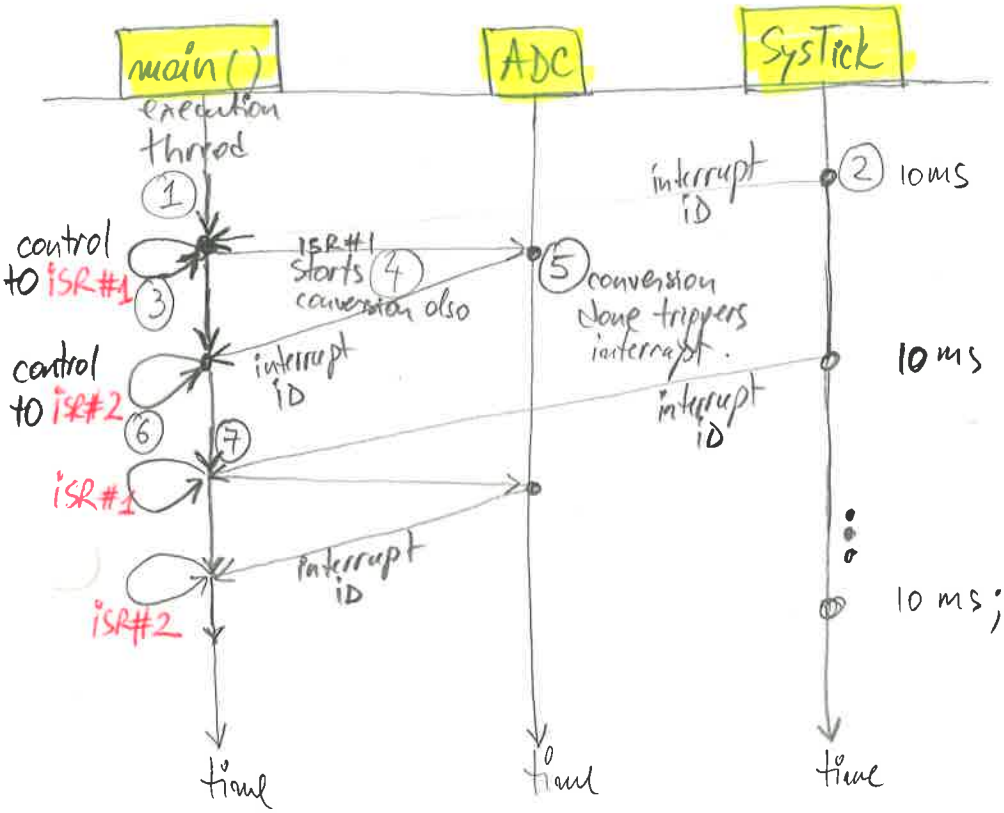
main program

ISR executed when AD conversion done

```
void ADC_IRQHandler() {
    AD_last = result of conversion
    AD_done = 1        ISR#2
}
```

ISR executed every other 10 ms

```
void SysTick_Handler( ) {
    if (ticks = 100) {
        clock_1s = 1;
        ticks = 0;
    }
    if (timetick++ >= (AD_last>>8)) {
        timetick = 0
        leds <<= 1                    ISR#1
        LED_out (leds)
    }
    ADC_StartCnv( )
}
```



Questions:

- how often are the interrupts from ADC triggered?
- how often or how fast is the while loop inside main() executed? and how is that affected by execution of ISR#1 and ISR#2?
- how often (w/ what period) are the LED's turned on/off from right to left?
- if SysTick would be generating interrupts every other say 2 sec, how would the LED's control be to human eye? would calibration help?
- how often is the LCD refreshed w/ new information?
- is SER_putString() (inside main()) affecting performance?

Note: Discussion Based on [Blinky2] example from lab #2.

```
void ADC_init(void){

    LPC_SC->PCONP |= (1<<12)|(1<<15); //enable power to ADC & IOCON
                                      //(1)
    // page 63 of user manual.

                                               //      bits 19:18
    LPC_PINCON->PINSEL1 &= ~(3<<18);  // 111...10011...11 ≡ Mask
              ->PINSEL1 |= (1<<18);   // set bit 18 to '1', i.e.,
                                      // make pin P0.25 to
    //(3)                             // function as AD0.2
                                      // page [108] of user manual.
    LPC_PINCON->PINMODE1 &= ~(3<<18); // page 112
              ->PINMODE1 |= (2<<18);  // make "10" bits 19:18 ≡
                                      // pin P0.25 has no
                                      // pull up/down resistor!

    // Also part of (1):
    LPC_ADC->ADCR = (1<<2)|(4<<8)|(1<<21); //AD Control Register
                                           //select operation mode
    //                  15 14 13 12 11 10 9 8
    // 000...0100...00...00000100 00000100
    //        ↑            ↑            ↑
    //      bit 21       bit 10        bit 2
    //        ↑            ↑            ↑
    // AD converter        |     select which of AD0.7:0
    // is operational      |     pins is selected & converted
    //              Bits 15:8 ≡ 00000100 (Bin) = 4 (Dec.)
    //              Clock is divided by this value.
    // See pages 576, 577 of user manual.

    LPC_ADC->ADINTEN = (1<<8); // AD interrupt Enable register
    NVIC_EnableIRQ(ADC_IRQn); // (4)  //page 578
    // ADC end of conversion          //only individual
                                      channels will
                                      generate
                                      interrupts!
}
```