PM för laboration Lab 9, Interruptprogrammering, Hans Sundgren 2010-04-07

Jag har labbat med interrupt och pulsgivare för både PIC6F628, 690 och 886 som en del av mitt projekt. Allt är baserat på kursmaterialet som har varit bra hjälp (interrupt-materialet läste jag in tidigt då jag anade att jag behövde funktionen). En del smällar har jag gått på men nu funkar det mesta.

I mitt projekt fungerar följande ”nästan till 100%”.

PIC16F886: 2 interrupt

- 1 Hz interrupt baserad på Timer1 och 32.768 kHz klock-kristall
- Pulsgivare för att öka/minska datum- och tidparametrar för önskad tid. Detta görs med 1 Hz interruptet avstängt.

PIC16F690: 1 interrupt

- Pulsgivare för att styra digital potentiometer som ingång tilll förstärkarsteg. Dessutom byte av ljudsignal vid en snabb rotation upp/ner, dvs:
  - Upp-vridning, valfri hastighet: Öka volym
  - Ned-vridning, valfri hastighet: Minska volym
  - Snabb upp följd av snabb ner: Stega upp ljudmod ett steg, samt volymändring
  - Snabb ner följda av snabb upp: Volymändring

Kod för huvudprocessorns 2 interrupt:

```
interrupt int_server(void)
{
   int_save_registers                     // W, STATUS (and PCLATH)
   char sv_FSR = FSR;

   if( RBIF == 1 )                        // Rotary encoder interrupt
   {
      if (TRANS == 0)               // Button pressed
         {
            trans_mode = ! trans_mode;         // Start/stop signal
             delay10 (3);
            if (trans_mode == 1)
            {
               I_TRANS=1;
            }
         if (trans_mode == 0)
            {
               I_TRANS=0;
            }
         }

      transit.0 = ROTB;                   // Read rotary encoder value
      transit.1 = ROTA;
      if( transit == 0b00.01 )            // Compare value
      {
         down=1; //test
         LED0=1; delay250us(10); LED0=0;  // Blink LED
      }
      if( transit == 0b01.00 )
      {
```

```
      up=1; //test if  LCD is problem
      LED1=1; delay250us(10); LED1=0;  // Blink LED
   }
   transit.2 = transit.0;              // Replace old with new
   transit.3 = transit.1;
   RBIF = 0;                           // Reset interrupt flag
}

if (TMR1IF)                    // Timer 1 overflow interrupt
{
   TMR1IF = 0;                 // Reset overflow flag
   TMR1H = 0x80;               // Reset counter
   TMR1L = 0x00;
   if (start_mode == 0)
   {
   I_START=1; delay250us(30);  I_START=0;  // Blink Start button
      }
   if (start_mode ==1)
   {
      I_START=1;
      tick_pos++;              // Step up tick_pos one second (not needed )
      sec1++;                  // Step up one second
      update_clock ();         // Main clock updated every second

      if (tick_pos == 60)
      {
         tick_pos = 0;
      }
      if (tick_pos == 0)      // was '00'
      {
         set_bitvalue();       // Set all 60 bits with initial valu
         first=0;
         if (display_mode == 3)
         {
             update_display_mode_3 ();  //Update bits
         }
      }
      output_bitvalue();       // Binary pulse on output
      }
   if (display_mode == 1)
   {
      display_time ();
   }
   check_btn ();
}
FSR = sv_FSR;
int_restore_registers          // W, STATUS (and PCLATH)
}
```