```
;********************************************************
;* Laser pcb printer
;* For Attiny13
;* Mark Lerman
;* Version 2.00 - Start 1/26/2016
;* Uses manual feed clutch
;* Programmed via usb programmer
;* 9.6 MHz /8 fuse=1.2MHz clock
;* uses signal from MF Clutch as signal is printing
;* Using old narrow media sensor as pis
;* 16 inch carrier
;* Carrier feed manually initiated
;********************************************************

;ver 2.00
.Include "tn13def.inc"

.DEF rmp0       =R16    ;multipurpose register
.DEF ticks      =R17    ;interrupts counter`
.DEF secs       =R18    ;seconds counter

; bit definitions
.EQU mfps       =1              ;to manual feed paper sensor
.EQU pis        =4              ;to paper in sensor
.EQU es         =2              ;to exit sensor
.EQU sol        =3              ;mf solenoid clutch, lo when actiated by print command
.EQU npis       =0              ;optical pis




.CSEG
.Org $0000
;interrupt vectors are 1 byte in attiny13
        rjmp    start           ; Reset Handler
        reti                            ; IRQ0 Handler
        reti                            ; PCINT0 Handler
        reti                            ; Timer0 Overflow Handler
        reti                            ; EEPROM Ready Handler
        reti                            ; Analog Comparator Handler
        rjmp    timint          ; Timer0 CompareA Handler
        reti                            ; Timer0 CompareB Handler
        reti                            ; Watchdog Interrupt Handler
        reti                            ; ADC Conversion Handler
;

start:
        ldi rmp0, low(RAMEND) ; Set Stack Pointer to top of RAM
        out SPL,rmp0


; main program start
;initializes with mfps, pis and es hi
        rcall   set_ports               ;set up ports
        rcall   set_ticks               ;set up 25 msec interrupts
        sei                                     ;start interrupts
;******************************************************************************************


;do 30 second pause to allow printer to do initialization
;is 30 sec long

delay:
        clr             secs            ;seconds counter
del1:
        clr             ticks
del2:
        cpi             ticks,40 ;1 second?
        brne    del2            ;not yet
        inc             secs            ;bump seconds
        cpi             secs,30         ;30 seconds?
        brne    del1            ;no, keep going



;
;******************************************************************************************
;using mf clutch signal to tell that user has started printing
;new pis is through sensor, picks up hole in carrier
;new pis goes lo when activated


main:
        clr             ticks
main0:
        cpi             ticks,120       ;3 seconds?
        brne    main0           ;not yet
        cbi             portb,mfps      ;tell printer to move input rollers


main00:
        sbic    pinb,sol;loop till mf clutch pulled in (lo) when print starts
        rjmp    main00

;printer sometimes pulls Solenoid in "randomly", so ignore these
```

```
;if NPIS is lo - i.e. no carrier present

        sbis    pinb,npis       ;if carrier not present is false alarm
        rjmp    main00          ;so don't print yet

;**************************************************************************************
*******
;test time to see range after start of vclutch lo
;clutch is in for 4 seconds
;PIS must go lo between 3.875 and 4.575 seconds after clutch goes lo - .7 second window
;since travel is 3.873 IPS, max distance is 2.711 inches
;if 2 inches from starting point till hole in carrier, .258 sec/inch * 2 = .516 seconds
;stop beep at 3.875 -.375 sec = 3.5, giving extra time to push carrier into drum interface
;approx 1 sec window (1.075)

        clr ticks
        main000:
        cpi     ticks,140       ;3.5 seconds
        brne    main000
        sbi     ddrb,3  ;make output hi, beeper off


;loop till  npis goes lo
;npis sees hole through carrier
        cli                                     ;no interrupts
main00000:
        sbic    pinb,npis       ;loop till hole in carrier seen by npis=lo
        rjmp    main00000               ;



;16 inch carrier - 14" numbers work well, leave alone
;carrier is now moving, printer is getting ready to print
;actual start of physical printing is fixed time from PIS hi->lo
;must reset the 3 sensors at correct time,
;11" paper takes 2.84 seconds to traverse each sensor
;.258 sec/inch=3.873 inches/sec;
;7.07" pis to es * .257=1.825 sec /.025 =73 ints
;3.2" mfps to pis * .257=.8224 sec /.025 = 33 ints
;3.2" mfps to pis, so when pis goes lo, (14-3.2=10.8") left til mfps goes hi
;10.8*.258= 2.7864 sec = 111.4 ints

;14" * .258 sec/inch =3.612 sec=144.48 ints for 14" carrier to pass
;33 ints mfps to pis
;73 ints pis to es
;when pis goes lo is 10.8" to trailing edge carrier=111.4 ints
;when es goes lo is 3.73 inches between mfps and trailing edge of carrier
;3.73 inches*.258 ips = .96234 seconds /.025=38.49 interrupts
```

```
        cbi             portb,pis               ;pis to lo, we are printing
        sei                                             ;start timer
        clr             ticks                   ;start timing
main11:
        cpi             ticks,73        ;1.825 sec to reach exit sensor=7.07 inches
        brne    main11
        cbi             portb,es                ;exit sensor lo

;all sensors lo at this point
;now have to reset them
        clr             ticks
main12:
        cpi             ticks,38                ;3.73 inches
        brne    main12
        sbi             portb,mfps              ;mpfs reset

main13:
        cpi             ticks,71
        brne    main13
        sbi             portb,pis               ;reset pis
main14:
        cpi             ticks,144
        brne    main14
        sbi             portb,es                ;reset es

;after print done mf solenoid pulls in again stopping roller
;then releases again

        cbi             ddrb,3                  ;make input again
        rjmp    main                    ;and do again



;*************************

;25 msec interrupts
set_ticks:
        ldi     rmp0,0x02       ;ctc on ocr0a
        out     tccr0a,rmp0
        ldi     rmp0,0x04       ;prescale by 256, =.213.33msec/ct
        out     tccr0b,rmp0
        ldi     rmp0,117                ;117*.2133=24.9561 msec/int
        out     ocr0a,rmp0
        ldi     rmp0,0x04       ;0a output compare enabled
        out     timsk0,rmp0
        ret

;init ports - all sensors hi
;b0,b3 input, rest output
set_ports:
        ldi     rmp0,0x16       ;0 in, rest out
        out     ddrb,rmp0
        ldi     rmp0,0xfe       ;outputs hi, input pullup on
        out     portb,rmp0
        ret

Timint:
        inc     ticks
        reti
```