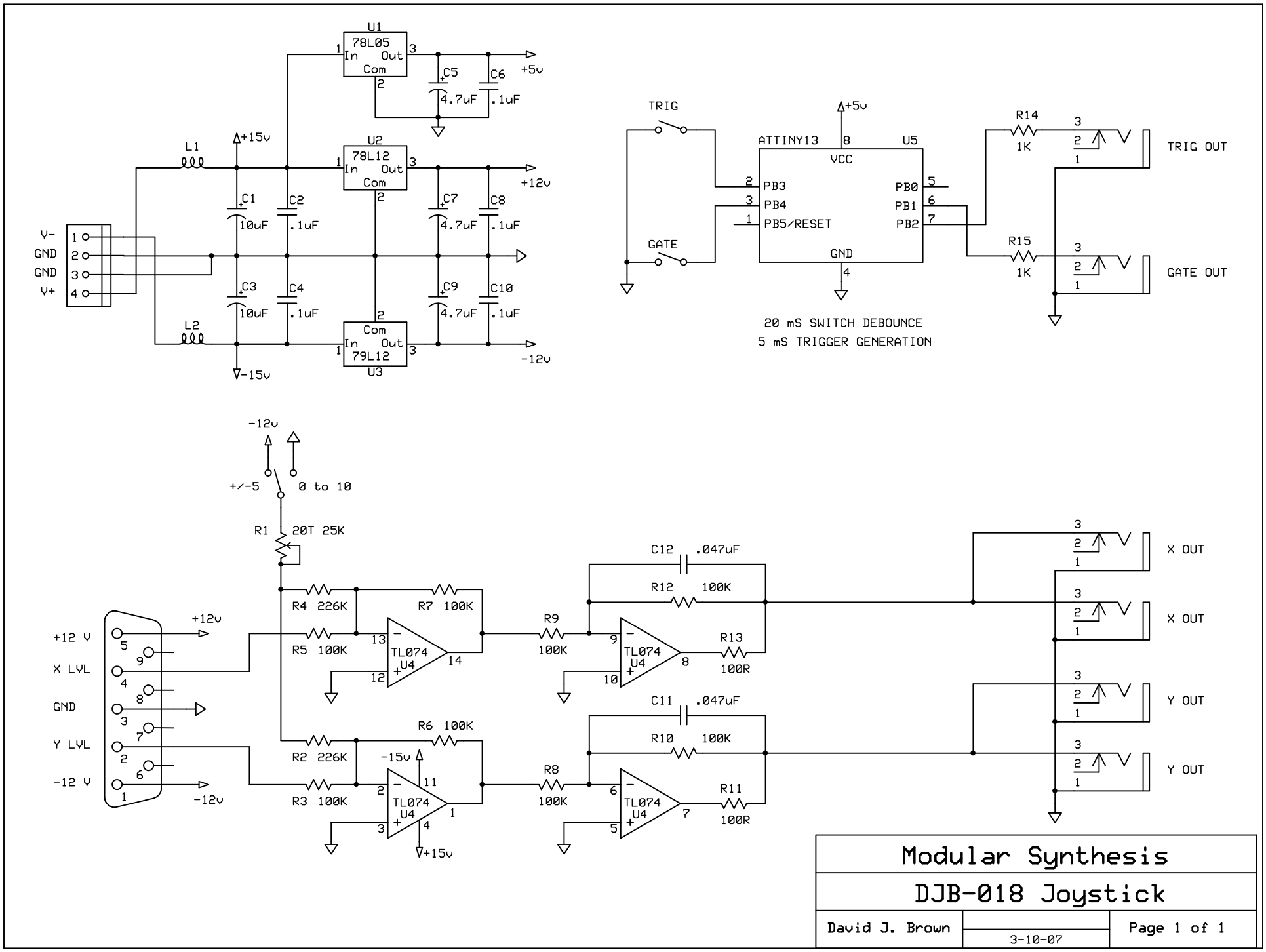
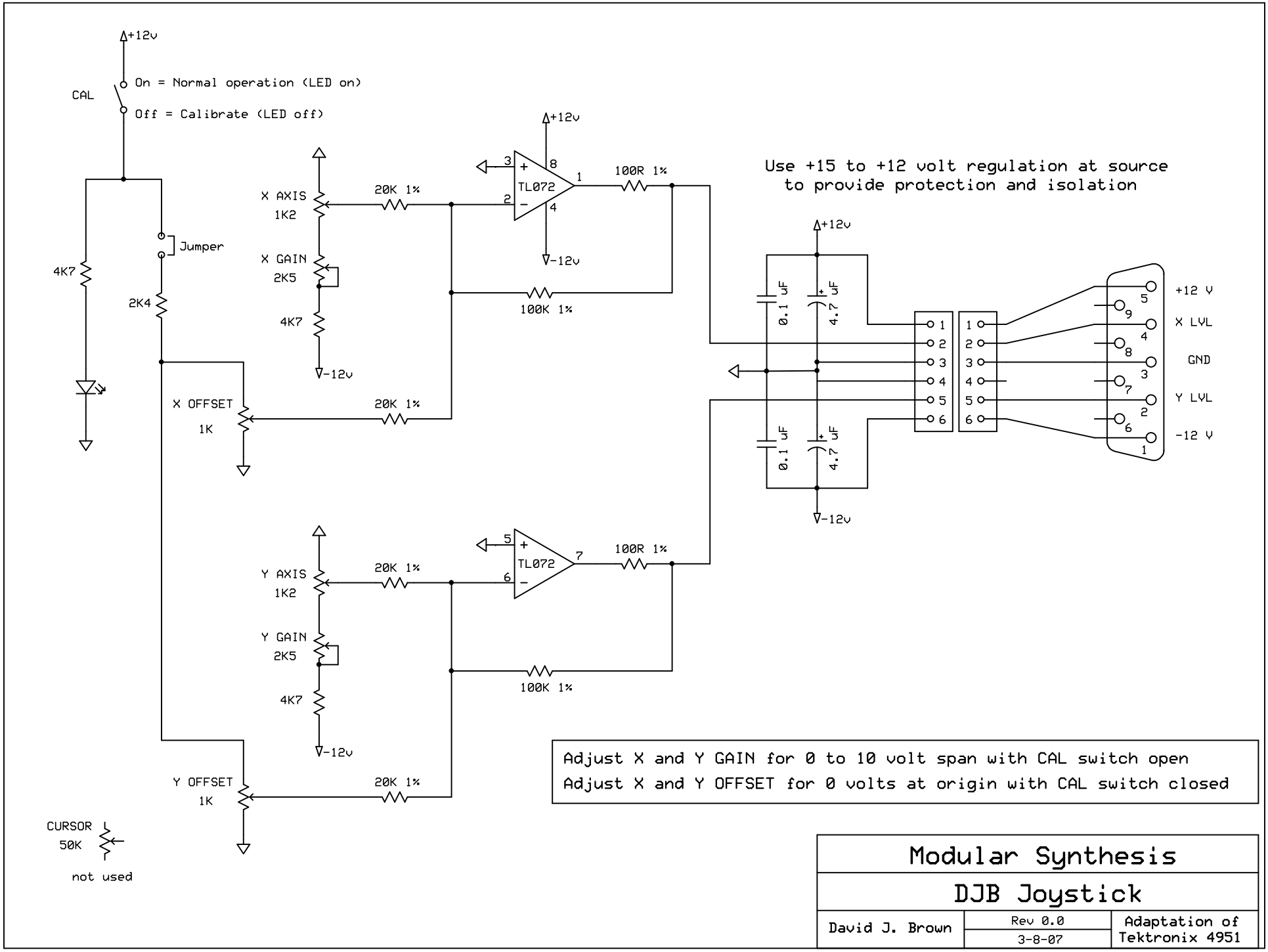

U1
78L05
In Out
Com
U2
78L12
In Out
Com
U3
79L12
Com
In Out
C1 10uF
C2 .1uF
C3 10uF
C4 .1uF
C5 4.7uF
C6 .1uF
C7 4.7uF
C8 .1uF
C9 4.7uF
C10 .1uF
L1
L2
+15v
-15v
+5v
+12v
-12v
V-
GND
GND
V+
TRIG
GATE
+5v
ATTINY13
U5
VCC
PB3
PB4
PB5/RESET
PB0
PB1
PB2
GND
R14 1K
R15 1K
TRIG OUT
GATE OUT
20 mS SWITCH DEBOUNCE
5 mS TRIGGER GENERATION
-12v
+/-5
0 to 10
R1 20T 25K
R4 226K
R7 100K
R5 100K
R2 226K
R6 100K
R3 100K
TL074 U4
R9 100K
R8 100K
C12 .047uF
R12 100K
R13 100R
C11 .047uF
R10 100K
R11 100R
+12 V
X LVL
GND
Y LVL
-12 V
X OUT
X OUT
Y OUT
Y OUT
Modular Synthesis
DJB-018 Joystick
David J. Brown
3-10-07
Page 1 of 1


+12v
CAL
On = Normal operation (LED on)
Off = Calibrate (LED off)
Jumper
4K7
2K4
X AXIS
1K2
X GAIN
2K5
4K7
-12v
20K 1%
TL072
100R 1%
100K 1%
X OFFSET
1K
Y AXIS
1K2
Y GAIN
2K5
Y OFFSET
1K
CURSOR
50K
not used
Use +15 to +12 volt regulation at source
to provide protection and isolation
0.1 uF
4.7 uF
+12 V
X LVL
GND
Y LVL
-12 V
Adjust X and Y GAIN for 0 to 10 volt span with CAL switch open
Adjust X and Y OFFSET for 0 volts at origin with CAL switch closed
Modular Synthesis
DJB Joystick
David J. Brown
Rev 0.0
3-8-07
Adaptation of
Tektronix 4951

```
;-----------------------------------------------------------------------------------------;
; LICENSE AGREEMENT:                                                                      ;
; This program is free software. You can redistribute it and/or modify it under the terms of   ;
; the GNU General Public License as published by the Free Software Foundation, either version  ;
; 2 of the License, or (at your option) any later version. No part of this code may be used    ;
; in any commercial application without prior written permission.                         ;
;                                                                                         ;
; This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY,    ;
; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.    ;
; See the GNU General Public License for more details.                                    ;
;                                                                                         ;
; Copyright (c) March 10, 2007  David J. Brown                                        ;
;                                                                                         ;
;-----------------------------------------------------------------------------------------;
;
;TriggerGen.asm
; developed for the ATTINY13
;
; written by David J. Brown
; email: davebr@earthlink.net
; web: http://modularsynthesis.com
;
; created: March 10, 2007
; last modified: March 10, 2007
; revision 0.0
;
; rev 0.0 initial code release
;
;------------------------
; specifications
;------------------------
;
; This program debounces a trigger and gate switch with a 20 mS
; scan and generates a gate signal and a 5 mS trigger pulse.
;
;------------------------
; hardware
;------------------------
;
; pb0           not used
; pb1           gate out
; pb2           trigger out
; pb3           trigger switch
; pb4           gate switch
; pb5           reset-0
;
.equ    trig_in =3                  ;pin 2
.equ    trig_out=2                  ;pin 7
.equ    gate_in =4                  ;pin 3
.equ    gate_out=1                  ;pin 6
```

```
;
; The internal rc oscillator is used with a +5 volt regulated
; power supply.
;
;------------------------
; fuse selections
;------------------------
;
; Brown-out detection disabled; [BODLEVEL=11]
; Int. RC Osc. 4.8 MHz; Start-up time: 14 CK + 64 mS [KCKSEL=01 SUT=10]
;
;------------------------
; registers
;------------------------
;
;               =r0                 ;r0 not used
;               =r1                 ;r1 not used
;               =r2                 ;r2 not used
;               =r3                 ;r3 not used
;               =r4                 ;r4 not used
;               =r5                 ;r5 not used
;               =r6                 ;r6 not used
;               =r7                 ;r7 not used
;               =r8                 ;r8 not used
;               =r9                 ;r9 not used
;               =r10                ;r10 not used
;               =r11                ;r11 not used
;               =r12                ;r12 not used
;               =r13                ;r13 not used
;               =r14                ;r14 not used
;               =r15                ;r15 not used
.def tempr      =r16                ;r16 temp register
;               =r17                ;r17 not used
;               =r18                ;r18 not used
;               =r19                ;r19 not used
;               =r20                ;r20 not used
;               =r21                ;r21 not used
;               =r22                ;r22 not used
.def delay_ms   =r23                ;r23 number of mS to delay
.def delay_l    =r24                ;r24 delay counter
.def delay_h    =r25                ;r25 delay counter
;               =r26                ;r26/xl not used
;               =r27                ;r27/xh not used
;               =r28                ;r28/yl not used
;               =r29                ;r29/yh not used
;               =r30                ;r30/zl not used
;               =r31                ;r31/zh not used
;
;------------------------
; assembler directives
```

```
;------------------------
;
.nolist
.include "tn13def.inc"
.list
.listmac
;
;------------------------
; ram
;------------------------
;
.dseg
.org    sram_start
;
;
;------------------------
; interrupt vectors
;------------------------
;
    .cseg
    .org 0x000
;
    rjmp start                      ;0x000  reset
    rjmp irq_none                   ;0x001  int0
    rjmp irq_none                   ;0x002  pcint0
    rjmp irq_none                   ;0x003  timer0 ovf
    rjmp irq_none                   ;0x004  ee rdy
    rjmp irq_none                   ;0x005  analog comp
    rjmp irq_none                   ;0x006  timer0 comp-a
    rjmp irq_none                   ;0x007  timer0 comp-b
    rjmp irq_none                   ;0x008  wdt ovf
    rjmp irq_none                   ;0x009  adc conversion
;
irq_none:
    reti
;
;------------------------
; main program
;------------------------
;
start:
    ;initialize stack and pins
    ldi tempr, low(ramend)          ;initialize stack
    out spl, tempr
    ldi tempr,0b00000110            ;0=input
    out ddrb,tempr                  ;set directions
    ldi tempr,0b11111001            ;1=pullup
    out portb,tempr                 ;set pullups
    cbi portb,trig_out              ;reset trigger output
    cbi portb,gate_out              ;reset gate output
```

```
    ;stop watchdog
    wdr                             ;reset watchdog timer
    in tempr,mcusr
    andi tempr,~(1<<wdrf)           ;clear wdrf
    out mcusr,tempr
    in tempr,wdtcr                  ;get prescalar settings
    ori tempr,(1<<wdce)|(1<<wde)    ;keep settings to prevent unintentional time out
    out wdtcr,tempr
    ldi tempr,(0<<wde)              ;turn off watch dog timer
    out wdtcr,tempr
;main loop when trigger switch is open
trig_open:
    rcall check_gate                ;check and process gate
    sbic pinb,trig_in               ;check if trigger switch closed
    rjmp trig_open
    sbi portb,trig_out              ;set trigger out
    ldi delay_ms,5                  ;5 mS trigger pulse width
    rcall wait_ms
    cbi portb,trig_out              ;reset trigger out
;main loop when trigger switch is closed
trig_closed:
    rcall check_gate                ;check and process gate
    sbic pinb,trig_in               ;check trigger switch
    rjmp trig_open                  ;open
    rjmp trig_closed                ;still closed
;
;debounce delay and process gate switch
check_gate:
    ldi delay_ms,20                 ;delay 20 mS to debounce switch
    rcall wait_ms
    sbis pinb,gate_in               ;check if gate switch closed
    rjmp set_gate
    cbi portb,gate_out              ;reset gate out
    ret
;
set_gate:
    sbi portb,gate_out              ;set gate out
    ret
;
;------------------------
; utility subroutines
;------------------------
;
;wait delay_ms mS @ 4.8 MHz RC OSC at +5 volts
;registers used
; delay_ms=number of mS to delay
; delay_h:delay_l=delay time counter
wait_ms:
    ldi delay_h,0xfb
    ldi delay_l,0xc7
```

```
wt_1:
    adiw delay_h:delay_l,1
    brne wt_1
    dec delay_ms                    ;decrement mS count
    brne wait_ms                    ;loop until done
    ret
;
;------------------------
; end of program
;------------------------
```