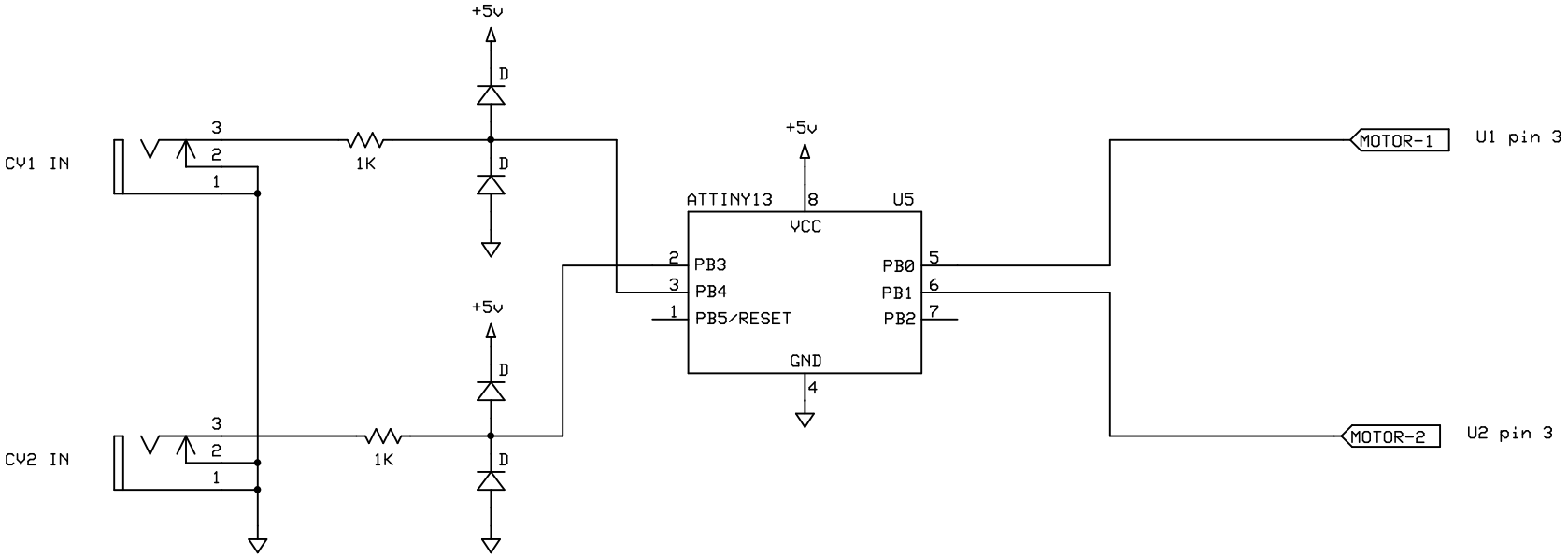

36.5 Hz MOTOR CONTROL FOR RAMSEY ELECTRONICS LLS1 MODULE

0 to 5V CV ADUSTS 50% to 100% DUTY CYCLE

Modular Synthesis		
LLS1 Motor Control		
David J. Brown	3-28-07	Page 1 of 1

```
;-----------------------------------------------------------------------------------------;
; 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 28, 2007  David J. Brown                                            ;
;                                                                                         ;
;-----------------------------------------------------------------------------------------;
;
; MotorGen.asm
; developed for the ATTINY13
;
; written by David J. Brown
; email: davebr@earthlink.net
; web: http://modularsynthesis.com
;
; created: March 28, 2007
; last modified: March 28, 2007
; revision 0.0
;
; rev 0.0 initial code release
;
;------------------------
; specifications
;------------------------
;
; This program converts two 0 to 5 volt control signals into PWM outputs
; for controlling laser motors. An output frequency of 36.5 Hz was chosen
; for best motor control.
;
;------------------------
; hardware
;------------------------
;
; pb0           output A
; pb1           output B
; pb2           not used
; pb3           control voltage B in
; pb4           control voltage A in
; pb5           reset-0
;
; 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 [CKSEL=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
;               =r23                ;r23 not used
;               =r24                ;r24 not used
;               =r25                ;r25 not used
;               =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,0b00000011            ;0=input
    out ddrb,tempr                  ;set directions
    ldi tempr,0b11111100            ;1=pullup
    out portb,tempr                 ;set pullups
    ;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
    ;initialize timer for PWM
    ldi tempr,0b10100001            ;phase correct PWM, clear A & B on upcount
    out tccr0a,tempr
    ldi tempr,0b00000100            ;/256 prescale for 36.5 Hz frequency
    out tccr0b,tempr
    ldi tempr,128                   ;default 50% duty cycle
    out ocr0a,tempr                 ;set compare A match
    out ocr0b,tempr                 ;set compare B match
    ;initialize ADC
    ldi tempr,0b00100010            ;Vcc reference, left adjusted, adin2
    out admux,tempr
    ldi tempr,0b10000110            ;ACD on, /64 prescale
    out adcsra,tempr
get_duty:
    ;get control voltage A and set output A
    cbi admux,0                     ;select adin2
    sbi adcsra,6                    ;start conversion
wait_2:
    sbic adcsra,6                   ;check conversion complete
    rjmp wait_2
    in tempr,adch                   ;get top 8 bits
    lsr tempr                       ;reduce to 0 - 127
    sbr tempr,$80                   ;offset to 128 - 255
    out ocr0a,tempr                 ;set compare A match duty cycle
    ;get control voltage B and set output B
    sbi admux,0                     ;select adin3
    sbi adcsra,6                    ;start conversion
wait_3:
    sbic adcsra,6                   ;check conversion complete
    rjmp wait_3
    in tempr,adch                   ;get top 8 bits
    lsr tempr                       ;reduce to 0 - 127
    sbr tempr,$80                   ;offset to 128 - 255
    out ocr0b,tempr                 ;set compare B match duty cycle
    rjmp get_duty
;
;------------------------
; end of program
;------------------------
```