Application Note 5

Solid State Circuit Breakers

by Brenda Kovacevic

Introduction

Until very recently, few alternatives to electromechanical and magnetic circuit breakers existed. Designers were forced to live with such undesirable characteristics as arcing and switch bounce (with corresponding noise and wear), while accomodating large unwieldly packages in their high power systems.

Solid state technology applied to this traditional device has resulted in circuit breakers free from arcing and switch bounce, that offer correspondingly higher reliability and longer lifetimes as well as faster switching times. A typical solid state circuit breaker will switch in a matter of microseconds, as opposed to milliseconds or even seconds for a mechanical version.

New solid state products currently on the market utilize the many benefits associated with power MOSFETs to deliver a product far superior to earlier silicon versions. Power MOSFETs offer low on resistances (as compared to bipolar transistors), low voltage drops, low EMI, faster switching times and good thermal stability of key parameters.

However, two key advantages that the electromechanical devices have over the solid state versions are simplicity and low cost. For example, a simple commercial circuit breaker relay combination will sell for $4.00 to $6.00 in low volume . The existing solid state circuit breakers will run from several times that amount, and typically include many bells and whistles that the average designer can do without. This cost difference is somewhat less in military versions, as the mechanical devices must also undergo extensive testing.

One reason for the corresponding complexity of the silicon based systems is the power MOSFET drive circuitry required. If N-channel FETs are to be used (N-channel FETs are preferable to P-channel as they have roughly 2.5 times lower R_{DS} (On) and correspondingly lower cost), a charge pump or voltage tripler must be supplied to provide sufficient gate enhancement to turn on the FET. This involves supplying an oscillator as well as the necessary diodes and capacitors, which definitely take board/hybrid package space.

A simple, inexpensive solid state circuit breaker can be made using the MIC5013 power MOSFET predriver with overcurrent sense. This predriver was designed for driving N-channel FETs, and has an on-board charge pump to provide sufficient gate enhancement. This eliminates the issue of providing this enhancement externally; providing a one component solution to what once consumed extensive "real estate".

As any size FET can be driven by the MIC5013, almost any load can be accomodated. High inrush or inductive loads are driven with equal ease, greatly expanding the realm of possibilities for these circuit breaker topologies.

An internal comparator is used to sense an over-current condition; this feature allows the use of this product as a circuit breaker that can be programmed to trip at a specified current via choice of an external sense resistor. An overcurrent flag provides this information externally, allowing easy digital interface /control of the device. This feature allows its use in more complex, remotely controlled designs such as those currently used in high reliability applications.

Using this highly versatile device, four circuit breaker configurations have been devised; a low parts count, low cost externally resettable version, a minimal parts count remotely resettable version with indicator, a minimal parts count automatically resettable version, and a full blown power controller design with Z8™ microcontroller interface. Typical applications for the first three versions include a variety of commercial, industrial and military applications, such as battery pack circuit breakers/current limiting, electric vehicles, and heavy machinery. The latter design is useful in high end applications such as military avionics or industrial automation. It offers a substantial cost savings over the currently available remotely controllable electromechanical units, as well as most currently available hybrid designs of this complexity.

Minimum Parts Count Configuration

Figure 1 illustrates the most basic configuration. The overcurrent trip point is set via the design equations in this figure. The current sense operates via a comparator which compares the voltage on the sense pin to an offset version of the voltage on the source pin. The current on the threshold pin, set by choice of R_{TH}, is mirrored and returned to the source by a 1 kΩ resistor.

This sets the trip voltage of the comparator. When a fault condition occurs, an internal current sense latch is set, which turns off the power FET. The control input pin must be toggled low then high by the reset switch before the FET will be switched on again (after the short has been removed). A 330kΩ resistor is provided to hold the input low and keep the FET off until the circuit is reset. Advantages of this topology are its simplicity and correspondingly low cost.

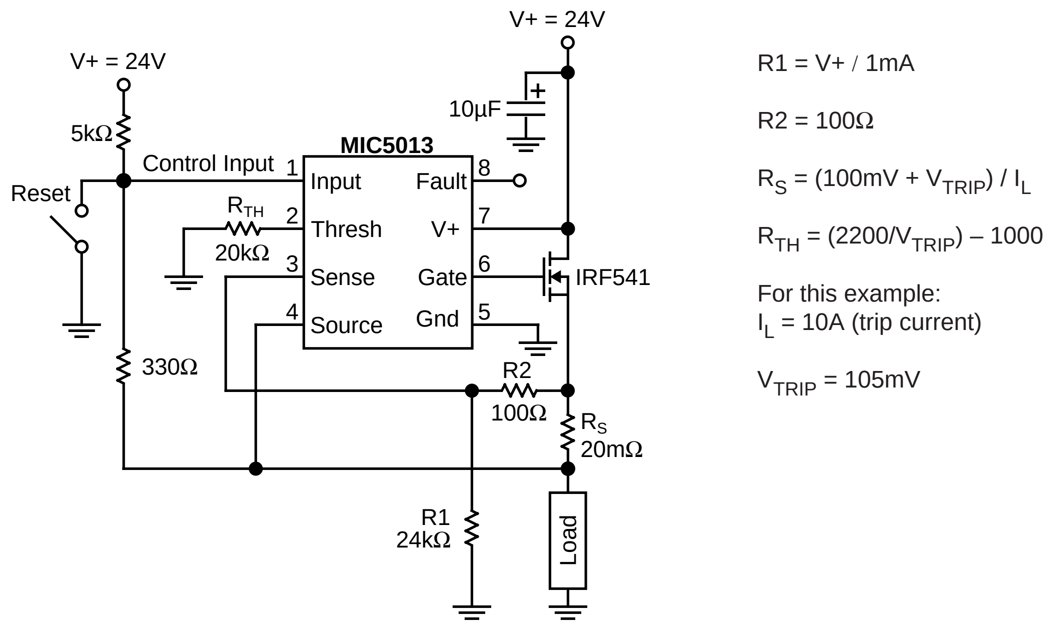

Figure 1: Basic Circuit Breaker/Switch Configuration

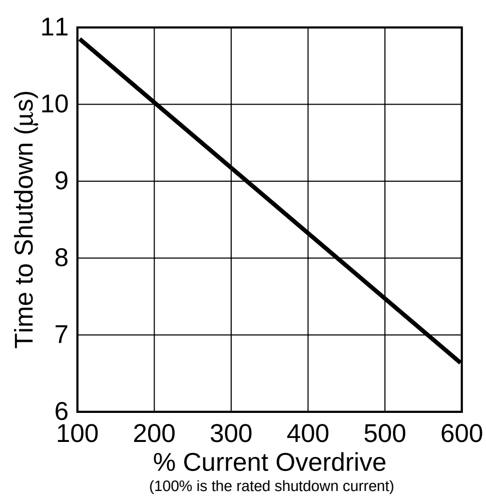

Figure 2: Shutdown Time vs. % Current Overdrive

Response Time

Figure 2 illustrates an advantage that is common to all MIC5013 based topologies: fast response times. A graph of shutdown time versus current overdrive is shown. The data was taken using this simple topology without the 330kΩ pulldown resistor, however, all configurations (with similar loads) will have a similar response as it is mostly a function of device parameters. (Note: This data was averaged from a small sample size; about 5-10% variation from this line may occur).

Response times in the order of µs means that a short circuit can be detected in time to prevent extensive damage, and is an improvement of an order of magnitude over electromechanical circuit breakers.

Remotely Resettable Configuration

The circuit breaker configuration of Figure 3 is designed to be used for applications requiring remote indication and reset capability. When the breaker is tripped, the fault output pin switches high (to a diode drop below the positive rail). This output is used to drive a remotely located LED. (If an incandescent lamp is desired, the fault output should be used to drive a power FET switch that could withstand the inrush generated). Resetting of the breaker is accomplished by toggling the control input with a remotely located switch. If the distance between the control point and the breaker is large, an optocoupler is recommended to open any ground loops that may occur. Many switch manufacturers offer a package that combines both the switch and the indicator while providing internal isolation, making this circuit even more compact. Shown here is the NKK-SS12SDP2-LE, a small slide switch suitable for instrument or control panels where space is at a premium.

Potential applications for this circuit include use as remotely controlled circuit breakers in aircraft with the indicator/switch located in the cockpit, industrial control panels, heavy machinery, and robotics.

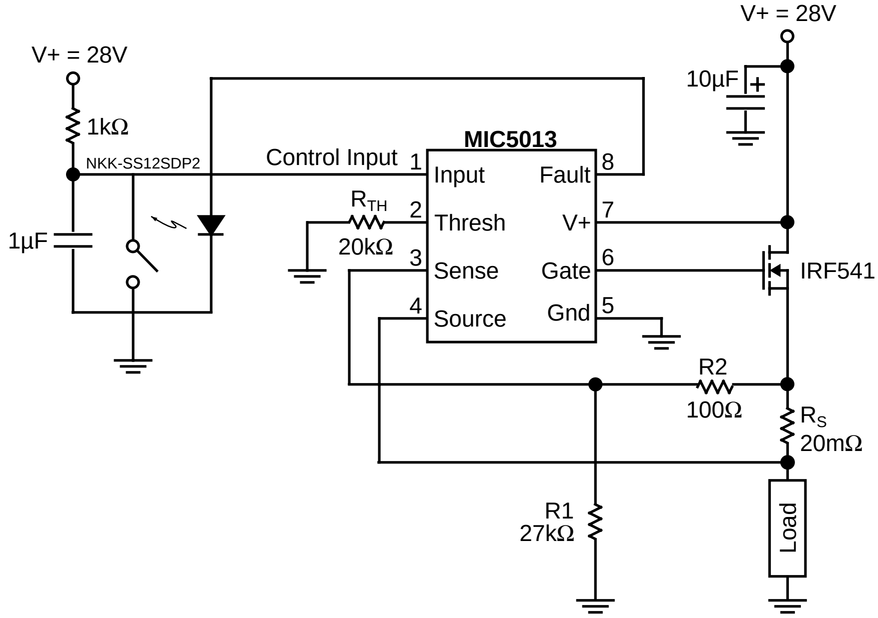

Figure 3: Remotely Resettable Circuit Breaker

Automatically Resettable Configuration

The third circuit,shown in Figure 4, is useful when automatic resetting is desired. This is accomplished by adding feedback from the fault pin back to the control input. A simple Miller integrator circuit is used to test the load every 18 ms until the short is removed. When the short condition no longer exists, the circuit latches on and operates as before. Although no reset button is necessary, an indicator could be added to the fault line if remote notification of a short circuit condition is desired.

The beauty of this configuration is that no human intervention is necessary once a short has occured. A possible drawback is that the gate does briefly turn on every 18ms to test the load. However, if the short still exists, it shuts down again in 10µs. This time duration is short enough to be acceptable in most applications.

Potential applications for this circuit include industrial automation, automotive circuitry, motor drive (stall sensing), and protection for power supplies/battery packs.

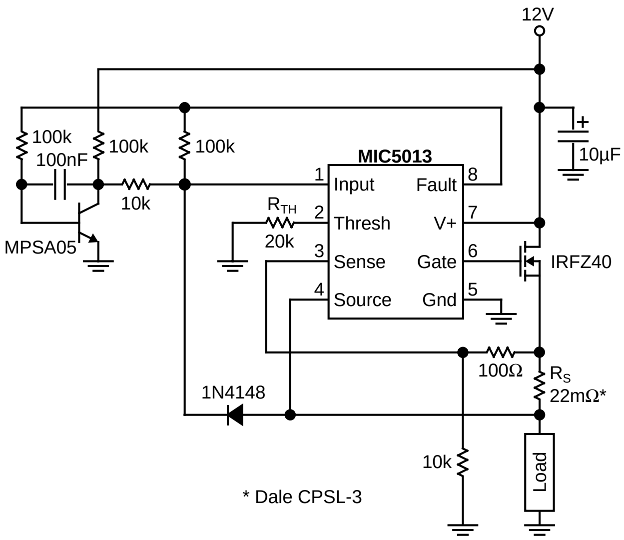

Figure 4: Automatically Resettable 10 A Circuit Breaker

Microcontroller Based Power Controller

A current trend in power electronics is the combination of intelligent power circuitry with microcontrollers; a so called "brains and brawn" combination. The power circuitry provides, in this case, the high current drive and circuit breaker function. The microcontroller can be used to make decisions in the event of a short, i.e., it can drive a warning signal, shut down other components of the system, or switch in a reserve or auxiliary motor (or pump, fan, heater, etc.).

An example of a microcontroller based power controller designed and built using the MIC5013 is shown in Figure 5. Here, three functions are monitored by the microcontroller; condition of the power supply (low or off), open load, and shorted load. If any of these three conditions exist, power is taken from the load and the control input of the MIC5013 and an appropriate LED is turned on. An additional LED is used to flag a hardware fault when an impossible condition (such as an open and short load) are flagged to the microcontroller.

Under normal operation (no fault condition exists), the microcontroller provides drive to the MIC5013 control input, and keeps bit 4 on I/O port 2 (P24) low, supplying drive to an LED signifying that conditions are "OK". (Note: a buffer may be necessary, as the MIC5013 is not TTL compatible).

The circuit breaker subsystem operates similarly to the other cases described earlier, however, all resetting is accomplished by the microcontroller. When the fault output goes high, indicating a short circuit has occurred, one input of the NOR gate is pulled high, causing a low output on the NOR gate. This toggles P32 (bit 2, port 3,) low, initiating the cond_init subroutine (see Figure 6 for Z8 code). This subroutine scans P20-P22 to determine which flag caused the NOR gate to go low. Upon determining that it was P20, P35 is brought low, providing the necessary toggling of the MIC5013 control input such that operation can resume once the short is removed (The MIC5013 current sense comparator output is connected to an internal latch which must be reset). Power has already been removed from the gate output of the MIC5013 by its internal current sense mechanism, shutting down the power FET and corresponding load. P26 is pulled low, lighting an LED that signifies that a fault has occurred.

When the fault is removed, the Z8 will restore power to the "OK" LED, shut down the "Overcurrent" LED, and restore power to the control input of the MIC5013. No isolation between the microcontroller and the MIC5013 was deemed necessary in this case, as the fault output is current limited by the voltage divider resistors, and tends to be fairly clean.

Open load detection is accomplished via the use of an LM301 op amp configured as a comparator. The LM301 was chosen for this application as it has more headroom than most op amps. The inverting input of the LM301 is set to 25 mV below the positive rail, which the non-inverting input will never reach unless the load is removed. The output of the op amp/comparator is fed to the HCPL-2602 optocoupler with the enable pin tied high. Under normal conditions, the output of the HCPL-2602 will be low; it toggles high in the event of an open load condition. The HCPL-2602 is also used to provide isolation between the digital and analog portions of the circuit. A high output from the HCPL-2602 causes the NOR gate to switch low, triggering the cond_int subroutine. The microcontroller reacts as before, removing power from the MIC5013 control input, and flagging the user that a problem has occurred.

The 1000 pF capacitor placed between the inverting and non-inverting inputs of the LM301 along with the 100 kΩ resistor serves as a noise filter, which prevents oscillations. Another way of doing this is to provide a small amount of hysteresis from the output back to the non-inverting input (See reference 4).

Low power detection is accomplished via the use of an optocoupler, the HCPL-3700, that also contains a Schmidt trigger. This provides hysteresis, allowing us to shut the system down when power reaches roughly 50% of rated value, and not turn back on again until we are at roughly 75% of rated value (These levels are chosen via selection of input resistor values and can be changed to meet the requirements of most systems. See the Hewlett-Packard Optoelectronics Designer's Manual for more details). Again, the optoisolator also provides isolation between the digital and analog portions of the circuit.

Shutdown and resetting of the system in the case of a low power condition is accomplished as before, by triggering the cond_int subroutine, which in turn scans port 2 to find the appropriate cause for the trigger and lights the corresponding LED.

If subroutine cond_int detects an impossible combination of conditions, i.e. short and open, a hardware fault has probably occurred. The microcontroller then lights an indicator LED attached to P34, and hangs up until the problem is removed.

The emergency override feature allows a pilot (or vehicle commander) to keep the system alive even though a short circuit has been detected. In a combat or other emergency situation, the equipment could be kept operating until the short circuit causes the FET to blow.

A switch located in the cockpit is used to provide this function. When it is depressed, IRQ2 (P31) is pulled low, causing the internal timer/counter to begin an 11 ms switch debounce count. If IRQ2 is still low (switch is still depressed) after 11 ms, then internal interrupt IRQ5 is activated on time out. Interrupt service routine T1_int then keeps power flowing to the control input of the MIC5013 , and toggles P23 high. This turns on the base of Q1, which pulls the signal on the sense input of the MIC5013 to ground, disabling the current sense function of the part. (If a 14-pin MIC5010 is used instead of the MIC5013, an external inhibit pin is available).

A key advantage of this circuit is that 2/4 interrupt lines and one complete I/O port is left unused. This would allow the microcontroller to be used for other functions in addition to power management.

If this is to be a dedicated power management system and the unused I/O has no other potential purpose, then some ideas for modifications include using an alphanumeric display instead of indicator LEDs, and including a self test mode with indicators on power-up.

If a PWM'ed load is to be used, the Z8 can be used to provide a variable frequency, variable pulse width signal by using the internal counter/timer registers (See the Z8 Design Manual for details). In this case, P36 should be connected to the control input of the MIC5013 instead of P35, and switch debounce will have to be performed in hardware instead of firmware. The MIC5013 can be switched up to a maximum frequency of 20kHz. Digital closed loop motion control can also be performed using the controller.

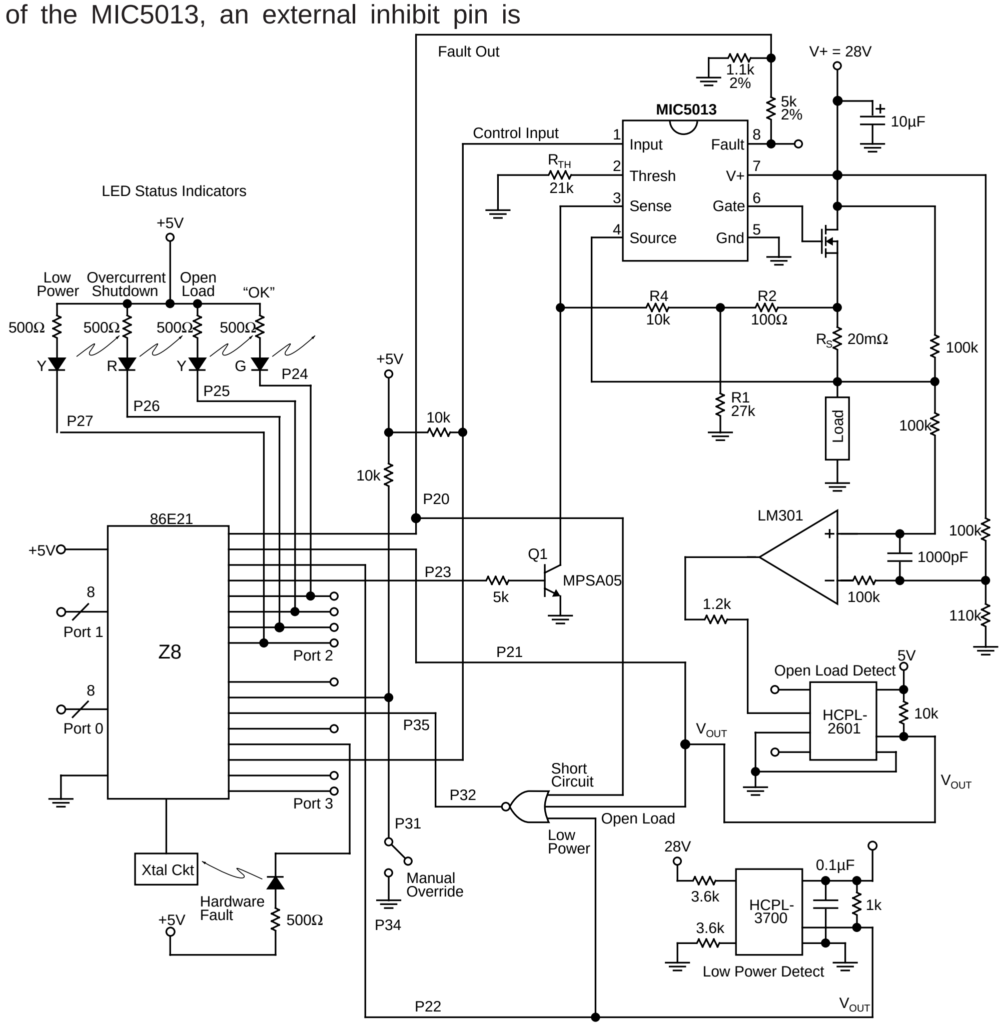

Figure 5: Z8 Based Power Controller

Summary

The MIC5013 MOSFET predriver with over current protection brings a whole new dimension to the world of power management with its versatility, ease of use, and quick response times. Four different lab tested circuit breaker configurations were presented and discussed; a minimum parts count version, a remotely resettable version, an automatically resettable version, and a complete microcontroller based power management system. Many more unique configurations are possible; a configuration to fit most needs can potentially be designed using the MIC5013.

References

1. The Z8 Design Manual, Zilog, 1985

2. The Optoelectronics Applications Manual, HP Optoelectronics, McGraw-Hill, 1981

3. Micrel Databook, 1995

4. Pease, R. A. , Troubleshooting Analog Circuits , Butterworth - Heinemann, 1991

5. Faber, Al and Kennelly, Bob, "Hybrid Power Controller Outperforms Conventional Circuit Breakers", *PCIM ,* November 1990, pg. 40

6. HP Application Note 1004, "Threshold Sensing For Industrial Control Systems With the HCPL-3700 Interface Optocoupler"

7. Frank, Randy and Psaenich, Al "Surviving Short Circuits", *Machine Design* , March 8,1990, pg 89

8. Conner, Margery, "Devices Let Aircraft Use Higher Voltages", *EDN ,* August 17, 1989, pg 59

9. asmS8™ Super 8/Z8™ Cross Assembler User's Guide, Zilog 1985

Figure 6: Z8 Microcode

```
.title CIRBR.S
.page 53                                ; set maximum lines/page to 55
.title                      CIRCUIT BREAKER CODE
;************************************************************************************
;           TITLE                P32.S
;           PROGRAMMER:          BRENDA KOVACEVIC
;           PURPOSE: THE FOLLOWING PROGRAM ENABLES THE Z8
;             MICROCONTROLLER TO RECEIVE DIAGNOSTIC INFORMATION
;             FROM A  SOLID STATE POWER CONTROLLER AND FEED THIS
;             INFORMATION BACK TO THE USER. FOUR INDICATIONS
;             ARE GIVEN; SHORT CIRCUIT, LOW POWER, OPEN LOAD,
;             AND 'OK' CONDITIONS ARE FLAGGED VIA THE USE OF LEDS
;             DRIVEN DIRECTLY BY THE Z8.
;
;************************************************************************************

;************************************************************************************
;                         EQUATES AND VARIABLES
;************************************************************************************

            dbnce_actv: .EQU     R0         ; working register r0 is the '
                                            ; 'debounce timer active' flag

;************************************************************************************
            .BEGIN
            .ORG        %8400
            int0:       jp       null_iret  ; unused interrupt
            int1:       jp       null_iret  ; unused interrupt
            int2:       jp       null_iret  ; unused interrupt
            int3:       jp       null_iret  ; unused interrupt
            int4:       jp       null_iret  ; unused interrupt.
            int5:       jp       T1_int     ; Counter/Timer 1 interrupt.

; First user-available location in RAM is at %8500
            .ORG %8500

start:
            jp          init                ; jump around ascii data,
                                            ; strings,...

.ascii 'created 2/26/91 by BLK.'

init:
; 1) Set up interrupts:  Interrupts are configured here.

            di                              ;
            clr         imr                 ; mask out all interrupts
            clr         irq                 ; clear out any pending
                                            ; interrupts
            ei                              ; initialize interrupt request
                                            ; enable latch.
            di
            ld          IPR,#00001000b      ; irq5 has highest priority

            ld          IMR,#00100000b      ; enables interrupt 5(internal
                                            ; timer interrupt);masks off
                                            ; unused interrupts

;  2) Initialize Register pointer and stack:
            srp         #%50                ; put scratch "working register"
                                            ; set at %50-%60
            ld          SPH,#%A0
            ld          SPL,#%00            ; top of external memory is the
                                            ; top of the stack
;  3) Initialize I/O Ports:

            ld          P01M,#11010011b     ; port 0 address and data, port 1
                                            ; output, external stack, normal
                                            ; timing
            ld          P2M,#00000111b      ; P20-P22 inputs; P23-P27
                                            ; outputs
            ld          P3M,#01000000b      ; Port 2 pullups open drain,P30-
                                            ; P33 int. inputs, P34-P37
                                            ; outputs  : P31 = Tin
;  4) Initialize Counter/Timers.

            ld          PRE1,#10000010b     ; set prescaler to 64 (decimal),
                                            ; single pass
            ld          T1,#10000000b       ; loads 256 in the timer, allows
                                            ; 11 ms count
            ld          TMR,#00101100b      ; load and enable t1, triggered
                                            ; internal clock mode
```

```
; 5) Initialize flag.
        clr   dbnce_actv                  ; start with a clean debounce
                                          ; timer flag

; 6) All set up. Enable interrupts and go!
        ei                                ; enable interrupts

status_check:
        tm    P3,#00000100b               ; check for bad condition
        jr    z,chk_pwr_cond              ; active low

good_status:
        ld    p2,#11100111b               ; sends power to 'OK' LED
        ld    p3,#00110000b               ; sends power to control input
                                          ; of MIC5013
        jr    ovrd_chk                    ; jump over subroutine call

chk_pwr_cond:
        call  cond_int                    ; check power circuits
ovrd_chk:
        tm    dbnce_actv,#1               ; If the emergency override has
                                          ; already been pressed, skip the
                                          ; test for emer. override.
        jr    nz,status_check             ; go back and start status check
                                          ; over the timer's already
                                          ; running
        tm    P3,#00000010b               ; Has the emergency override
                                          ; (P31) been pressed?
        jr    z,emer_ovrd                 ; If yes, trigger debounce timer

        jr    status_check                ; no - start over again looking
                                          ; for status
emer_ovrd:
        or    dbnce_actv,#1               ; set debounce timer active flag
                                          ; to indicate that the timer's
                                          ; rolling.
        or    TMR,#00100011b              ; start debounce timer rolling
        jr    status_check                ; continue to wait for something
                                          ; else to happen


;*****************************************************************************************************
;                    *****************************
;                    Subroutine cond_int
;                    ****************************
;*****************************************************************************************************
;Subroutine: P32 low , signals power malfunction
;Function:   Is tripped for any of the three malfunction conditions;
;Action      Subroutine cond_int reads port 2 to determine which
;            condition exists, and toggles the appropriate diagnostic bits
;             of port 2 or 3.
;*****************************************************************************************************
cond_int:

short_test:
        tm    P2,#00000001b               ; see if P20 is high (short
                                          ; condition)
        jr    z,open_test                 ; jump if no short
        and   P3,#11011111b               ; reset bit 5 of P3 to shut down
                                          ; MIC5013
        ld    P2,#10110111b               ; reset P26 to turn on
                                          ;           overcurrent LED
                                          ; fall through to test open load
                                          ; condition, open and short
                                          ; coincident indicates h/w fault
open_test:
        tm    P2,#00000010b               ; see if P21 is high (open load
                                          ; condition).
        jr    z,low_test                  ; jump if no open load condition

        tm    P2,#00000001b               ; Do we also have a short
                                          ; condition (illegal)?
        jr    z,open_only                 ; Jump if not

        jr    hw_fault                    ; Catastrophic h/w failure -
                                          ; indicate separately.
open_only:
        and   P3,#11011111b               ; reset P35 to shut down the
                                          ; MIC5013
        ld    P2,#11010111b               ; reset P25 to turn on "Open
                                          ; Load" LED
                                          ; fall through to low voltage test
```

```
low_test:
        tm    P2,#00000100b               ; see if P22 is high (low power
                                          ; condition).
        jr    z,end_cond_int              ; jump if no low-voltage fault

        and   P3,#11011111b               ; reset P35 to shutdown the
                                          ; MIC5013
        ld    P2,#01110111b               ; reset P27 to turn on "low
                                          ; power" LED
        jr    end_cond_int                ; done with power condition
                                          ; tests
hw_fault:
        and   P3,#11001111b               ; reset P34 to turn on "h/w fault"
                                          ; LED - we have a circuit
                                          ; breaker malfunction - and
                                          ; turn off the MIC5013!
        or    P2,#00010000b               ; turn off the "OK" LED, we have
                                          ; a HW fault and things are
                                          ; NOT OK!!

end_cond_int:
        ret

;*****************************************************************************************************
;Interrupt:  Emergency Override Switch Timer Interrupt
;Function:  Keeps the MIC5013 alive while shorted in emergency situations.
;Action:     When the manual override switch is depressed, internal timer
;            T1 begins counting for 11 ms (see main). At the end of this debounce
;            routine, interrupt IRQ5 is asserted.  This takes priority over the cond_int
;            subroutine, and keeps the control input to the MIC5013 on while
;            disabling the current sense by pulling the sense pin
;             to ground through transistor Q1.
;*****************************************************************************************************
T1_int:
        di                                ; disable interrupts
        and   dbnce_actv,#0               ; reset 'debounce active' flag
        and   irq,#11011111b              ; Reset the interrupt source

        tm    P2,#00000001b               ; Don't take action if there is no
                                          ; short
        jr    nz, end_T1_int              ; Bail out.

        tm    P3,#00000010b               ; Check to see if override switch
                                          ; is still depressed
        jr    nz,end_T1_int               ; If not, then it was just noise
                                          ; triggered
                                          ; go back to main.
        or    P2,#00010000b               ; Sends power to Q1 to disable
                                          ; current sense
        or    P3,#00100000b               ; Makes sure the control input is
                                          ; still on
end_T1_int:

        ei
        iret



;*****************************************************************************************************
;Interrupt:   Null interrupt
;Function:    Intercept any spurious interrupts.
;Action:      None. Just return from the interrupt.
;*****************************************************************************************************
null_iret:
        and   irq,#00100000b              ; Reset any spurious pending
                                          ; interrupt.
        iret


        .END
```