Using the HT45R38 for Pan Detection in Induction Cookers

D/N: HA0135E

Introduction

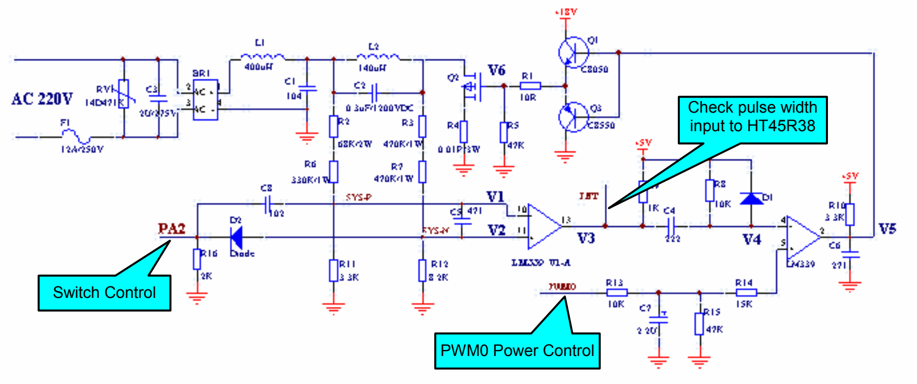
Induction cookers are now a readily available and popular domestic appliance. They possess the following advantages: no flame, no fumes, electrically efficient, easy to use etc. Induction cookers have evolved from their previous total hardware control to their present state where a microcontroller device is used for their control.

The Holtek A/D type MCUs are now extensively used in induction cooker applications, and offer the advantages of low cost, high efficiency, rapid development etc.

A wide selection of Holtek MCU devices are used in induction cooker applications including the HT46R47, HT46R22, HT46R23, HT46R12A, HT46R14A, HT46R32, HT46R34, HT46R322, HT46R342, HT45R38 etc.

A popular method of pan detection in induction cookers is current pan detection and pulse pan detection. In this application note the method of pulse pan detection is mentioned. Pulse pan detection has the following advantages: low power detection, pan detection distance is stable, high resolution, immune to input voltage variations, immune to the disk winding size influences and immune to resonant capacitor influences.

Induction Cooker Pan Detection Principle



Check pulse width input to HT45R38

Switch Control

PWM0 Power Control

Figure 2-1 Pan Detection Circuit

To monitor whether there is a load on the cooker, under conditions of no load (no pan) the IGBT will be off (set PA2 low). The oscillator formed by the coil L2 and resonant capacitor C3 also make a long time damped oscillator. Its oscillation period is about 40μs. Two synchronous signals (see Fig. 2-1 V1 and V2) alternatively change, making the comparator (see Fig. 2-1 U1-A) alternately switch. Finally, using the external interrupt line (see Fig.2-1 V3) to obtain more timing pulses. In the case of a load being present (a pan placed on the cooker), if PA2 is low, then the damping oscillation time of the oscillator formed by the oscillator circuit L2 and C3 is practically zero. (this indicates that the resonant total time is practically zero and will immediately stop oscillating, not that the resonant frequency is zero – its resonant period when compared with no load will be larger by about 45μs). Here the number of pulses on INT will be lower. The timing diagram is shown in Fig. 2-2.

With this principle, a high signal on PA2 will allow the IGBT to operate for a fixed period of time, at which time the system will start to could. After PA2 is set low and the IGBT stops operating, again the system timer again counts for a fixed time. Then the total count and the setup standard can be compared, the result of which can determine whether a pan is present on the cooker or not.

As the situation where the pan is removed, this is relatively simple. During the heating process, every 2 seconds, the heater will be turned off and a pan detection process carried out again. If a pan is detected then the heating process will continue according to the previous procedure, otherwise if no pan is detected continuously 10 times then the heating process will be terminated.

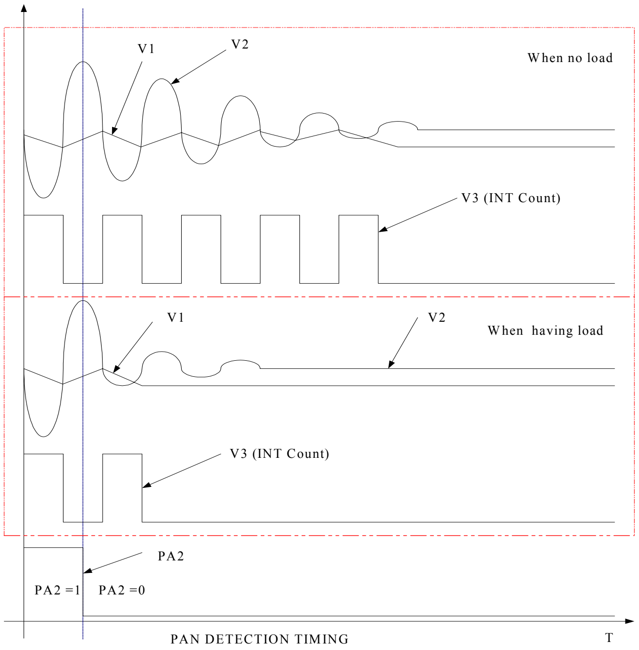


Fig. 2-2 Pan Detection Timing Wave Form

Pan Detection Software Design Description

The pan detection software design method is as follows: every 2 seconds a pan detection process will be executed. For each pan detection the PWM0 is set to 100ms. Then the cooker is switched on for 1ms, after 1ms the cooker is switched off and the number of pulses measured. A value greater than 18H will be taken as no pan detection, a value smaller than 18H will be taken as a pan detected. When a no pan detection is made continuously 10 times, then the cooker will automatically turn off.

Actual code is as follows:

```
;-----------pan_detection()----------------------
PAN_DETECTION:
   MOV  A, 10
   SUB  A, pan_count       ;check 10 times for no pan detection
   SZ   C
   JMP  PAN_COUNT_BLW10
   CALL RESET_ALL_DATA     ;If continuous 10 times no detection then RESET
                           ;related parameters
   CALL BEEP1              ;If continuous 10 times no detection then generate
                           ;warning signal
   MOV  A, 60                 ;Extend fan time extend 1 minute - switch off
   MOV  second_delay_fan, A
   MOV  A, 1
   MOV  display_state, A      ;Setup display condition
   MOV  A, 120
   MOV  minute_delay_poff, A   ;Setup automatic switch off time
   JMP  EXIT_PAN_DETECTION

PAN_COUNT_BLW10:
   SZ   second_delay_pan       ;Check every 2 sec. interval - here check if
                               ;2 sec. time has elapsed
   JMP  EXIT_PAN_DETECTION

PAN_DETECTION_TIME_ARRIVED:
   SNZ  err_check              ;If already have error condition then stop pan
                               ;detection
   JMP  $+3
   SZ   second_delay_err
   JMP  EXIT_PAN_DETECTION

   SZ   pan_detect_step        ;Pan detection has two execution steps, the
                               ;first is initialisation
   JMP  PAN_DETECTION_STEP_EQU1

   MOV  A, power_level         ;Determine power level, if power_level = 3,
                               ;then no need to setup PWM0 value
   SUB  A, 3
   SZ   C
   JMP  $+3
   MOV  A, 080h  ;Setup PWM0 value
   MOV  PWM0, A

   SET  PA0_START_PWM          ;Start PWM0 to execute pan detection

   CLR  int_count              ;Pan detect count value cleared

   MOV  A, 25                  ;Setup PWM o/p time value = 25*4ms=100ms
   MOV  pan_time, A

   MOV  A, 1                   ;change to following mode
   MOV  pan_detect_step, A
   JMP  EXIT_PAN_DETECTION

   PAN_DETECTION_STEP_EQU1:
   SZ   pan_time               ;check if PWM o/p time is reached
   JMP  EXIT_PAN_DETECTION

PAN_TRY_AGAIN:                 ;At this step the PWM is already working at
                               ;100ms
   CLR  ET0I
   CLR  EEI1
   CLR  ERCOCI                 ;Clear External OSC Interrupt。

   MOV  A, 50
   MOV  time_var, A            ;setup extension time (Delay
                               ;time=50*6+2=302us)
   SET  EEI0                   ;enable external 0 interrupt
   SET  PA2_ENABLE_PWM         ;start cooker
   CALL DELAY_XUS              ;int_count=12~14@26kHz
   CLR  EEI0
   CLR  PA2_ENABLE_PWM         ;Switch off

   MOV  A, 200
   MOV  time_var, A            ;setup delay time (Delay
                               ;time=200*6+3=1202us)
   SET  EEI0                   ;Enable external interrupt 0
   CALL DELAY_XUS              ;Delay if have pan, int_count+=1~3,
                               ;else int_count+=48~51
   CLR  EEI0                   ;Disable external interrupt 0

   SET  EEI1                   ;Enable external interrupt 1
   SET  ET0I                   ;Enable external interrupt 0
   SET  ERCOCI                 ;Set External OSC Interrupt

   SZ   int_count              ;so, when no pan,
                               ;int_count=12+12*4=60=3cH
   JMP  INT_COUNT_LG0          ;when have pan, int_count=14+3=17,
                               ;considerate error
                               ;add 4 again. so standard
                               ;value=18H(adjustable)
   CLR  PA0_START_PWM          ;Disable PWM0 output
   CALL BEEP                   ;Circuit malfunction – emit “beep” sound
   MOV  A, 08h                 ;Setup pan detection index value
   MOV  err_number, A          ;Circuit error case
   SET  err_check              ;Setup pan detection flag
   INC  pan_count              ;Pan detection plus 1
   JMP  INT_COUNT_ALNORMAL

INT_COUNT_LG0:
   MOV  A, 18h                 ;According to pulse count (int_count), decide
                               ;if pan is present
   SUB  A, int_count
   SZ   C
   JMP  INT_COUNT_BLWMAX

   SZ   pan_count              ;For first no pan detection, do not
                               ;immediately set flag – check again
   JMP  DETECTION_FAIL
   INC  pan_count
   CLR  int_count
   JMP  PAN_TRY_AGAIN          ;First no pan detection – check again

DETECTION_FAIL:
   CLR  PA0_START_PWM          ;Disable PWM0 output
   CALL BEEP
   CLR  flag_pan_detect        ;Clear no pan detect flag

   MOV  A, 1
   MOV  err_number, A          ;Set error code index - no pan

   SET  err_check              ;Setup setup error flag
   INC  pan_count              ;No pan – add 1, if continuous 20 counts, then
auto switch off cooker
   JMP  INT_COUNT_ALNORMAL

INT_COUNT_BLWMAX:              ;Here pan has been detected
   CLR  err_check              ;clear pan presence flag - have pan
   CLR  err_number             ;Clear error flag index
   CLR  pan_count
   SET  flag_pan_detect        ;set already checked pan presence flag

INT_COUNT_ALNORMAL:
   MOV  A, 2
   MOV  second_delay_pan, A    ;setup 2 sec. timer, check for pan presence
                               ;each interval
   CLR  pan_detect_step
EXIT_PAN_DETECTION:
   RET
;-----------end of pan_detection()---------------
```

Conclusion

This example has introduced the principles of using the HT45R38 for pan detection in induction cookers and it was explained how the actual program is used to implement this function. The detection principle is that during a period of time if no pan is detected to be on the cooker, then the cooker will be turned off.