AVR422; Phase Angle Control of Power Circuits

8-bit AVR® Microcontroller

Application Note

Features

- **0 - 100% Power Control**
- **Dimmer for Mains Light, Heater Control, Spot Welding Control Unit**
- **Minimum of External Components Required**
- **Suitable for any AVR MCU**
- **Example Code Included for AT90S8515**

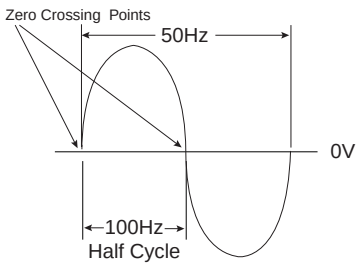
Introduction

This application note describes the process of power regulation by means of Direct Phase Angle control using an AVR microcontroller. This allows the designer to directly control mains power line applications and implements it using a minimum of external components.

Theory of Operation

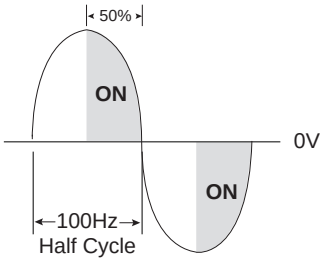
The mains power supply provides power at a fundamental frequency of 50Hz or 60Hz for each full cycle, this gives a period of 20ms at 50Hz or 16.6ms at 60Hz. In each cycle there is two zero crossing points, one at the start of the cycle and one at each half cycle. The time between these will be 10ms(100Hz) or 8.3ms(120Hz) respectively. See Figure 1.

Figure 1. 50 Hz AC Waveforms



If each half cycle can be further subdivided in this example by 100, then it is possible to control the main power that can be delivered to any mains powered load purely by choosing to switch on a triac (or thyristor pair) at the appropriate point in each half cycle. (e.g. for 50% main power the triac would be fired half way through each half cycle). See Figure 2.

Figure 2. 50% Power Setting

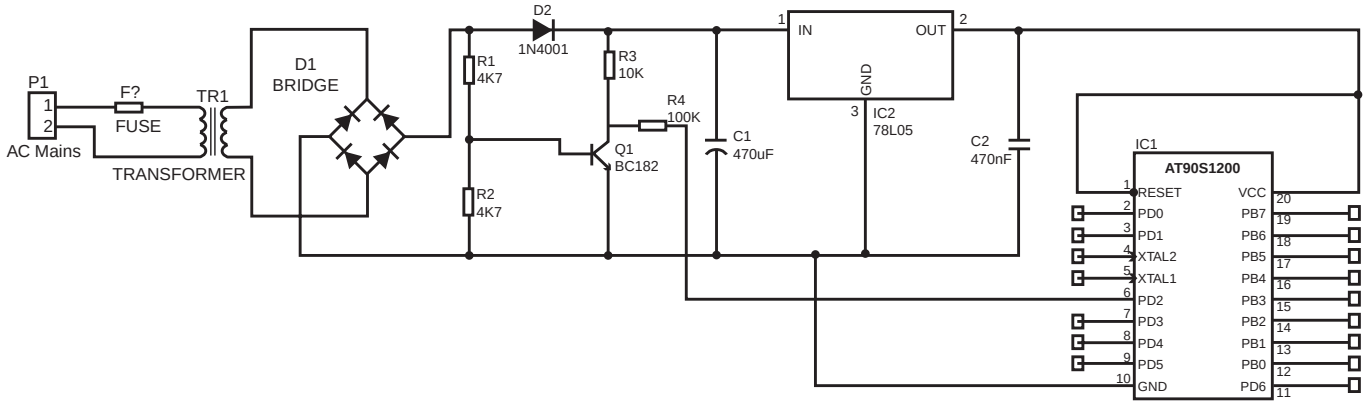


This method of control is particularly effective when controlling resistive loads (such as lamps or heaters), but also lends itself to control of welders and series motors and would provide 0 - 99% control. Because of the sinusiodional nature of the voltage, the power delivered to the load will not be proportional with the time-delay between the zero crossing point and the firing pulse. To achieve more accurate result use a look up table with sine values.

It works by providing a start point, which is normally the point at which the mains supply voltage passes through zero (The zero crossing point), and then count the number of sub-divisions until the required turn on point has been reached. At that point the triac is fired and the load is energized. The triac firing pulses are then removed at the next zero crossing point to prevent the triac from re-firing. The cycle then starts again.

Rev. xxxxA–02/99

Figure 3. Crossing Zero Detection



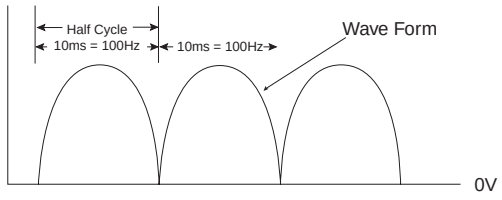
Hardware Description

The following hardware is designed to act as a simple digitally controlled light dimming unit but can easily be converted for lots of other purposes.

The schematic diagram (Figure 3) features the zero crossing detection circuit.

The circuit works by full wave rectifying an AC supply, the wave form at the anode side of D1 is a raw 100Hz output from the bridge and is not smoothed, See Figure 4.

Figure 4. Bridge Output

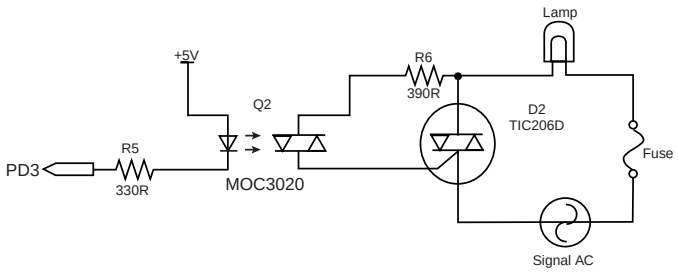


This wave form enables the two resistors R1/R2 to control the output of the heart of the circuit, Q1, which is biased hard on until the voltage on the AC output from the transformer/bridge drops to around 1.4 volts at which point it switches off, allowing R3 to pull the collector high thus creating a positive going pulse through the resistor R4 to Pin PD2 on the AVR.

The component values were chosen to work with a 24VAC transformer. The ratio of R1/R2 determine the pulse width. This pulse is used to generate an interrupt. This starts the power counter which in turn is used to determine the switching point for the power devices. The frequency of the counter is a multiple of the mains frequency so for 50Hz it will be 10Khz and for 60Hz it will be 12Khz. This can be derived simply by choooosing a crystal frequency of 6Mhz and setting the timer/counter to divide by 600 or 500 respectively.

Shown in Figure 5 is the switching unit used for mains control this circuit is controlling the power through a light bulb and thus its brightness (expected oscilloscope wave forms are shown in Figure 6).

Figure 5. Triac Connection



It works by pulsing PD3 Low (firing pulse generated by 10kHz counter) this in turn energises the opto coupler Q2 which triggers the triac D2 and allows the lamp circuit to energize for the set period (user power setting).

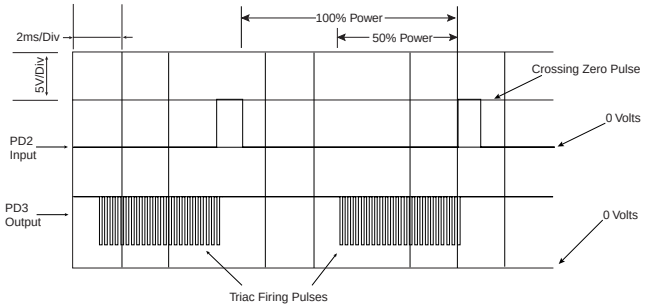
D2 - Triac TIC206D (Select part depending on current requirement).

Q2 - MOC3020 triac driver

R5 - Trigger current regulation for opto.

R6 - Current regulation for the gate of triac.

Figure 6. Expected Waveform Pattern at 50% Power



Software Description

There are 3 main routines in this system, two of these are interrupt service routines, Timer/counter 0 (TIMER0_OVF) and External interrupt 0 (INT0), the other routine is the initialisation routine (INIT) used to set up the stack, timer/counter, port and interrupts.

The register *Pwrcntr* holds the value for controlling the power setting in the TIM0_0VF routine. It is loaded after each crossing zero pulse with the current Power setting.

The register *Pwrstng* holds the value for user power setting and is loaded in the main routine with the user setting, this is then used to calculate the power setting.

"Init" - System Initialisation Routine

This routine is used to set up the timer/counter to either divide by 125 (for 60Hz) or 150 (for 50Hz). It also sets up the timer post scaler (/4) which is used to derive the 12Khz (for 60Hz) or 10Khz (for 50Hz) divider. The ports (PD2 - Crossing Zero Input, PD3 - Thyristor Firing Pulse Output) and interrupts are also set up using this routine.

"EXT_INT0" Interrupt Handling Routine

This routine clears the triac firing pulse enable flag T and the firing pulse output . It also derives the counter compare value (power setting). This is done by loading the R*sd* value previously worked out into the Rsd equate in the program. This is then copied into R17 (pwrcntr) where the user's power setting is subtracted R18 (prwstng) with the result being left in R17(pwrcntr).

"TIM0_0VF" Interrupt Handling Routine

This routine reloads the counter and and performs a test, t1 to see if the T flag has been set, If t1 is false the power setting is decremented and a further test, t2 is done to see if this setting has reached zero. If t2 is false the routine will carry out the return from interrupt instruction, if the t2 is true (i.e. power setting 0) then the T flag is set.

If t1 is true then the routine will jump to (fire_triac) where a pulse will be created.

"Main" Program Routine

This is where the user Power setting is loaded into R18 (pwrstng) in the example this is set to 50% i.e. 32h.

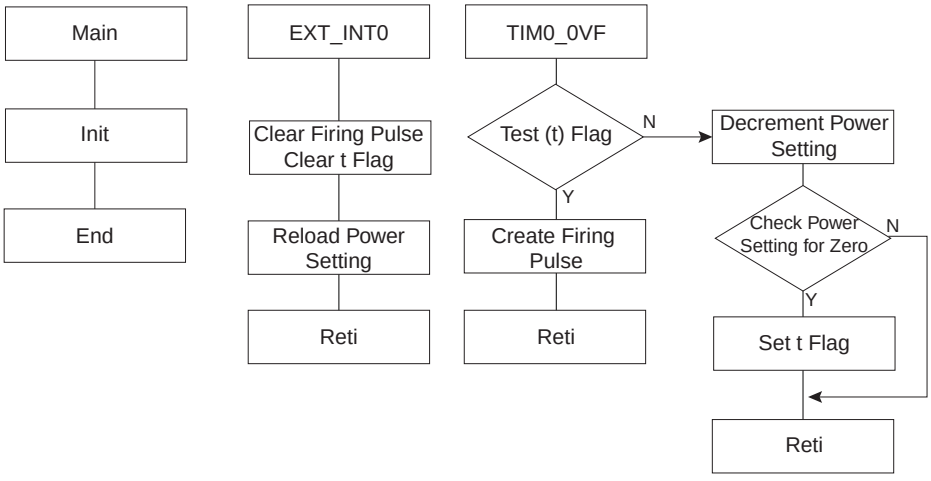
Modifying & Optimising

The code can be adapted to run at other Clock speeds, the control resolution can also be increase or decreased i.e. from 0.1% to 0.05%, these modifications are explained below.

Clock Speed

If this application is run on a faster or slower clock speed then the counter reload value will have to be recalculated, this is done as previously described earlier in the application with the result being loaded into Rv in the source code. If a different prescaler setting is used then the Value in Psk will need to be modified to suit the counter setup.

Figure 7. Phase Angle Control



Resolution

This also can be increased by recalculating the number of sub divisions as explained earlier in each half cycle. (i.e. the higher the number of subdivisions the higher the resolution of control). The result of which would be loaded into Rsd in the source code. Note if any change is made to the resolution then the counter reload value will also need to be recalculated.

The user setting will also be a subdivision of the Rsd value
e.g:

```
Rsd   =(64h)   100d     = 0 - 100d user setting where 100 = 100% Power
Rsd   =(C8h   ) 200d    = 0 - 200d user setting where 200 = 100% Power
Code size:              52 words, including Int Vectors
High Register Usage:    3
Status Flag Usage :     1 (T Flag)
Interrupt Usage:        External Interrupt 0
                        Timer/Counter0 Overflow interrupt
Pin Usage :             PD2 (INT0) for crossing Zero pulse I/P
                        PD3 for triac/ thyristor firing pulseO/P
```

Resources

Table 1. CPU and Memory Usage

Function	Code Size	Cycles	Example Register Usage	Interrupt	Description
Int Vectors	13 words	N/A	-	-	Interrupt vectors
Init	19 words	20	R16	-	Initialisation
EXT_INT0	5 words	9	R17, R18	INT0	Crossing Zero Detection Clear Firing Pulse Reload Power setting
TIM0_OVF	13 words	10 - 13	R16, R17	TIM0_OVF	10kHz Firing Pulse generator
main	2 words	3	R18	-	Enter Power Setting
TOTAL	52 words	-	R16, R17, R18	-	

Table 2. Peripheral Usage

Peripheral	Description	Interrupts Enabled
External Interrupt 0 (INT 0)	Crossing Zero Detection	INT0-Ext Interrupt 0 (Rising Edge Triggered)
Timer/Counter 0	10kHz Generator	TIM0_OVF Interrupt (CK/8 No Stop)
R16	General Scratch Pad	
R17	Power Setting Counter	
R18	User Power setting	
1 I/O pins	EXT_INT0	PD2 Input (External Interrupt 0 Pin)
1 I/O pins	Firing Pulses	PD3 Output

```
;**** A P P L I C A T I O N   N O T E   AVR422 ********************
;*
;* Title:             Phase Angle Control
;* Version:           1.0
;* Last updated:      98.05.14
;* Target:            AT90Sxxxx (All AVR Devices)
;*
;* Support E-mail:    avr@atmel.com
;*
;* Code Size:         52 words (including Int Vectors)
;* High Register Usage:3
;* Status Flag Usage: 1 (T flag)
;* Interrupt usage:   Timer/Counter0 overflow interrupt
;*                    External interrupt
;*
;* DESCRIPTION
;*
;* This application note shows how you can make a Phase Angle control
;* system using a minimum of external components.
;*
;* The system is completely interrupt driven which means that it will
;* run in the background. System timing is derived from a 4Mhz crystal.
;*
;* It works by creating a 10Khz (for 50Hz) or 12Khz (for 60Hz) counter and then counting
;* the number of pulses from a crossing zero pulse. Once the count value = the user setting
;* the system sets the firing enable flag (T) and then fires the triac by providing a series
;* of 5/6 Khz pulses until the next crossing zero is detected.
;*
;* The user power setting is stored in R18 and the value is updated at the end of each half
;* cycle of the mains input.
;*
;* It uses 3 registers all high
;* R16 - Scratchpad Register
;* R17 - Power Setting counter
;* R18 - User power setting register (0 - 99 in binary for 1% accuracy)
;*
;* It uses 2 port pins
;* PD2 (INT0) for crossing zero pulse input
;* PD3 for triac firing pulse output
;* *** Initialization
;*
;*  1. Set Stack Pointer
;*  2. Enable interrupts
;*  3. Set up Timer and postscaler values
;*
;* *** External Interrupt (x_zero)
;*
;*  1. Clear flag (T) (firing pulse enable flag)
;*  2. Clear Triac Firing pulse
```

```
;*  3. Derive power value from user input
;*
;* *** Timer Counter Interrupt (divide_gen)
;*
;*  1. Reload Counter (10/12kHz Generator)
;*  2. Branch to firing pulse generation if T flag is set (Firing pulse enable flag)
;*  3. Else check present point in mains cycle against counter setting

;***************************************************************************
    .include "8515def.inc"

  .def     Scratch = R16                        ;scratch Pad
  .def     Pwrcntr = R17                        ;Power Counter
  .def     Pwrstng= R18                         ;User Power Setting

  .equ     Rv  = $cd                            ;Counter Reload Value
  .equ     Rsd = $64                            ;Power Setting Binary adjust
  .equ     Psk = $02                            ;Counter/Prescaler Setting used in Init routine.

Reset:
    rjmp    Init
    rjmp    EXT_INT0                            ; External 0 interrupt  Vector
    reti                                        ; External 1 interrupt  Vector
    reti                                        ; Timer 1 Capture  Vector
    reti                                        ; Timer1 CompareA  Vector
    reti                                        ; Timer 1 CompareB  Vector
    reti                                        ; Timer 1 Overflow  Vector
    rjmp    TIM0_OVF                            ; Timer 0 Overflow interrupt  Vector
    reti                                        ; SPI  Vector
    reti                                        ; UART Receive  Vector
    reti                                        ; UDR Empty  Vector
    reti                                        ; UART Transmit  Vector
    reti                                        ; Analogue Comparator  Vector
    ; ******* Crossing Zero detection ********
EXT_INT0:
    clt                                         ; clear T flag (triac disable mode)
    sbi     portd,3                             ; Use this instead for use with Triac
    ldi     Pwrcntr,Rsd                         ; create binary power setting
    sub     Pwrcntr,Pwrstng                     ;Create power counter setting
    reti
    ; ******* 10Khz Generator and firing pulse generation ********
TIM0_OVF:
    ldi     scratch,Rv                          ; Reload value for 50Hz operation
    out     tcnt0,scratch                       ; reload counter with divider value
    brts    fire_triac                          ; triac firing
    dec     Pwrcntr                             ; check power setting
    brne    psd_skip                            ; if not at value then skip
    set                                         ; else set the T flag
psd_skip:
    reti
```

```
fire_triac:
    in      scratch,pind                        ; create 5Khz triac firing pulses
    sbrc    scratch,3
    cbi     portd,3
    sbrs    scratch,3
    sbi     portd,3
    reti
Init: ; ******* Stack Pointer Setup Code *******
    ldi     scratch, $02                        ; Stack Pointer Setup
    out     SPH,scratch                         ; Stack Pointer High Byte
    ldi     scratch, $5F                        ; Stack Pointer Setup
    out     SPL,scratch                         ; Stack Pointer Low Byte
    ; ******* Port Setup ********
    ldi     scratch, $FB                        ; I/O direction values
    out     DDRD, scratch                       ; Port D Direction Register
    ; ******* Timer0 Setup *******
    ldi     scratch, Psk                        ; Timer 0 setup
    out     TCCR0, scratch                      ; Timer 0 setup/Prescaler setup
    ldi     scratch,Rv                          ; Load counter with reload value
    out     tcnt0,scratch                       ; Load timer with 10Khz value for 50Hz operation
    ; ******* Interrupts Setup *********
    ldi     scratch, $40                        ; External Interrupt Enables
    out     GIMSK, scratch                      ; INT0 Enabled
    ldi     scratch, $03                        ; External Interrupt Rising Edge
    out     MCUCR, scratch                      ; INT0 - Low Level
    ldi     scratch, $02                        ; Timer Interrupt Enables
    out     TIMSK, scratch                      ; Timer 0: Overflow
    sei                                         ; Enable All Interrupts
    ;******* General House work *******
    clt                                         ; clear T flag (triac disable)
    sbi     portd,3                             ; clear firing pulse for Triac
Main:
    ldi     pwrstng,$32                         ;Load Power setting (50%)
    rjmp    main
```

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686677
FAX (44) 1276-686697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road
Tsimshatsui East
Kowloon, Hong Kong
TEL (852) 27219778
FAX (852) 27221369

Japan
Atmel Japan K.K.
Tonetsu Shinkawa Bldg., 9F
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4 42 53 60 00
FAX (33) 4 42 53 60 01

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

xxxxA–02/99/xM