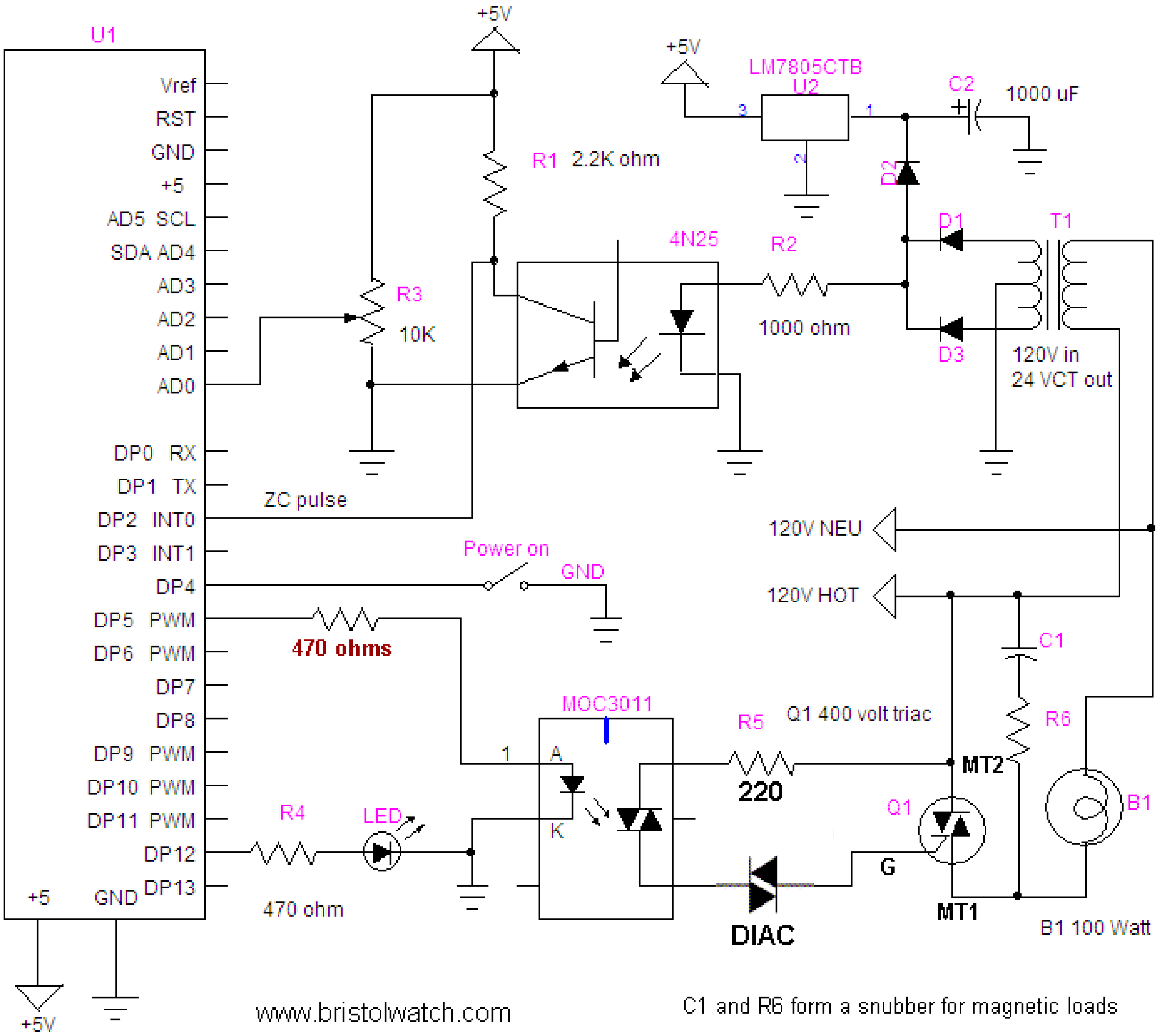
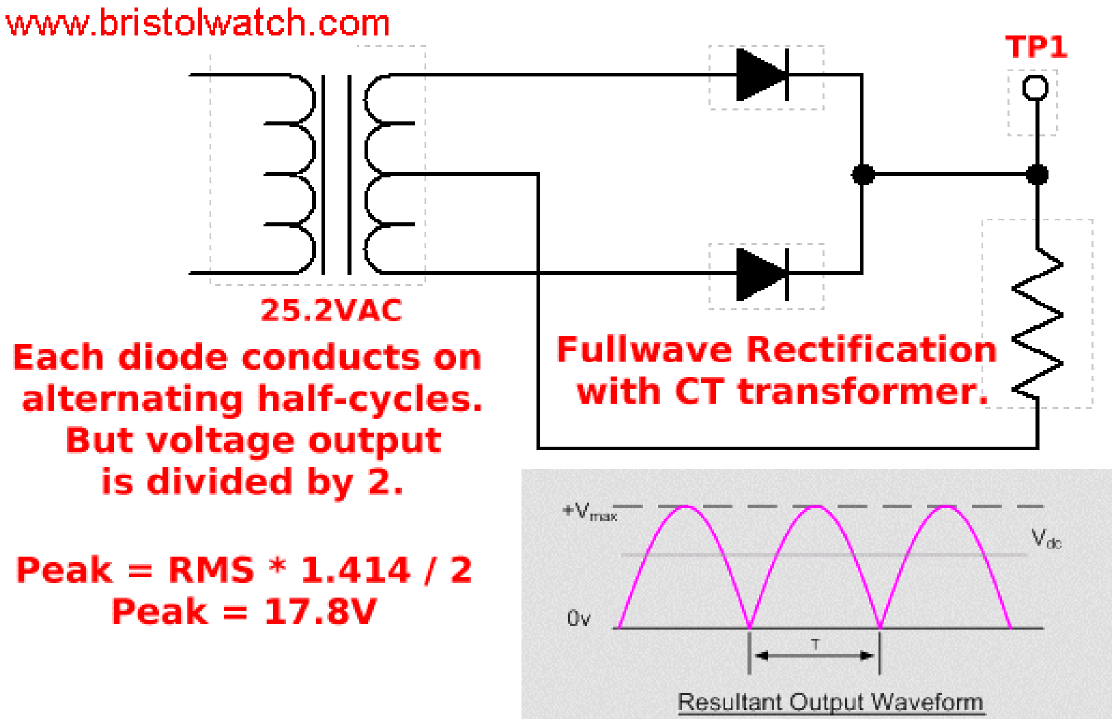


Basic schematic.

Arduino Hardware Interrupts Control AC Power

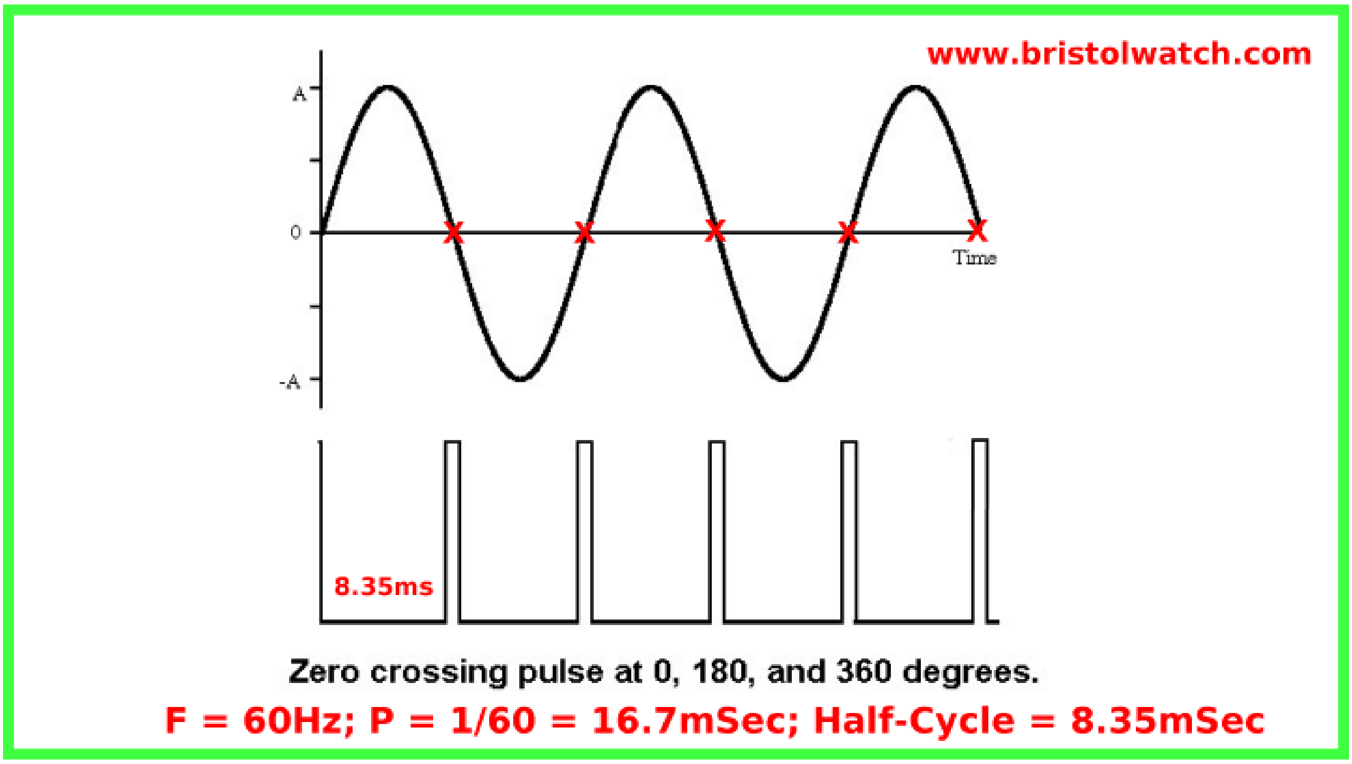
by Lewis Loflin

The purpose of this project is to demonstrate the use of a hardware interrupt to control AC power levels to a load such as a lamp or small AC motor. Note the "power on" switch must be pushed for the circuit to operate.



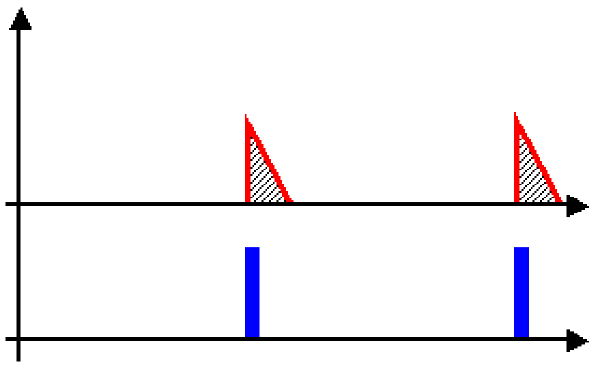
Full wave pulsating DC.

In the main circuit diagram above transformer T1, D1, and D3 produce a positive going pulsating DC with a peak voltage of about 18 volts and a frequency of 120 Hertz. Diode D2 blocks the filtering effect of capacitor C2, which with U2 supplies positive five volts for the microcontroller. See [Basic AC Rectification and Filtering]



Zero crossing pulse from 4N25 in relation to AC sine wave.

The 4N25 opto-coupler provides a narrow 120 Hertz pulse at zero and 180 degrees of the sine wave. This pulse is fed to digital pin 2 (Dp2) of the controller to trigger an interrupt when the sine wave passes zero and 180 degrees. (There's 360 degrees in a sine wave.) See [4N25 Opto-Coupler] (PDF file)

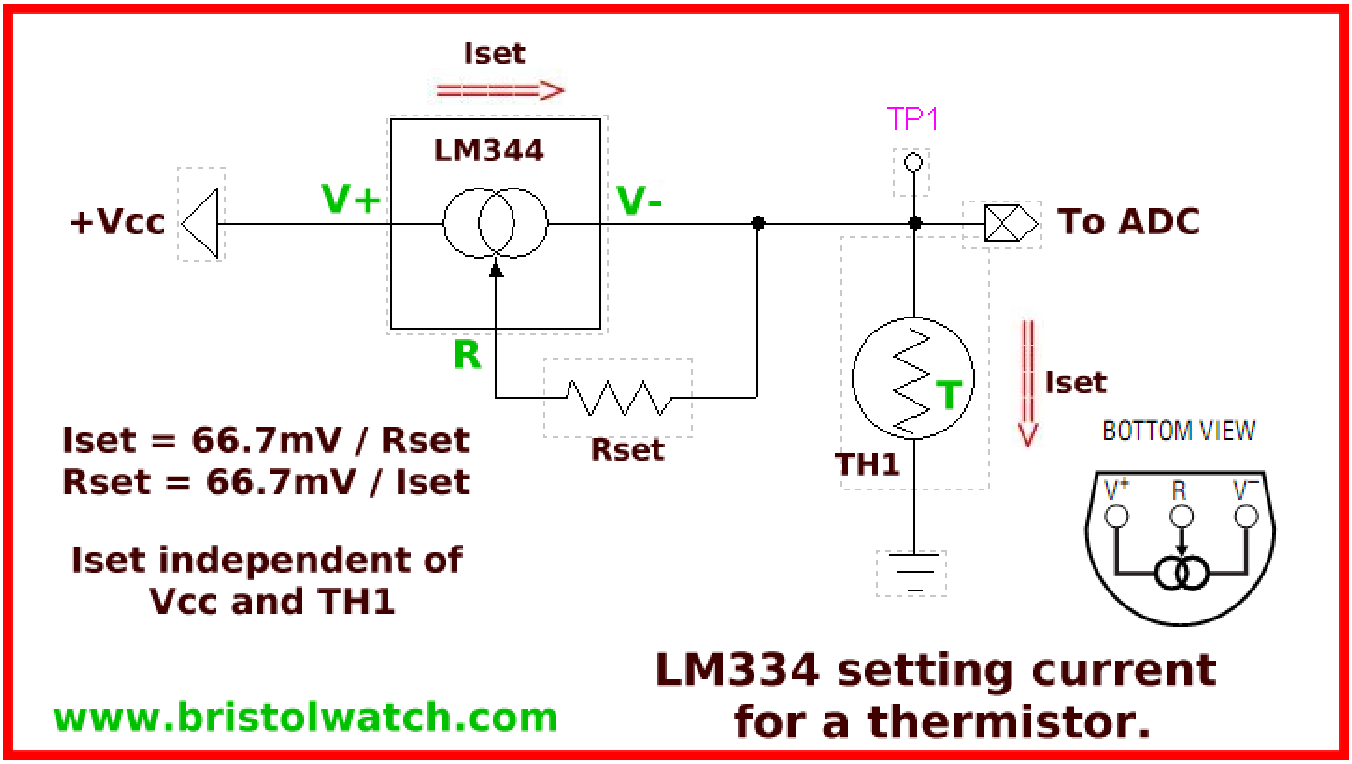


This illustrates to process with full-wave unfiltered D.C. but the process is identical with A.C.

By programming a delay of between .1 and 8.2 milliseconds based on the voltage value on pin Ad0 we control the firing point of a triac, transistor, or silicon controlled rectifier to control power output.

For this experiment one can use 24 volts A.C. and a 24 volt lamp. If the circuit is wired properly when the
control is adjusted the circuit should act as a lamp dimmer or speed control for a motor.
If the lamp dims as the control is adjusted clock-wise reverse the two outer connections on R3.

- Hardware Interrupts Tutorial for Arduino
- Basic Triacs and SCRs
- Solid State AC Relays with Triacs
- Light Activated Silicon Controlled Rectifier (LASCR)
- Arduino AC Power Control Using Interrupts
- In Depth Look at AC Power Control with Arduino



Sample circuit for photocell or thermistor connection to ADC (analog to digital converter) sensor pin.

Uses

LM334 CCS Circuits with Thermistors, Photocells YouTube
LM334 CCS Circuits with Thermistors, Photocells

A circuit such as this can do far more than a $3 lamp dimmer. By connecting a thermistor to measure temperature (or an appropriate thermocouple circuit) to one of the analog inputs we have a proportional heat controller saving energy costs. We could control a heating element or the speed of a blower fan proportional to temperature.

By connecting a photocell we could control light intensity of lamps in proportion to natural light, saving energy costs with say sky lights. I use a version of this on my water heater to control a narrow temperature range.

In a proportional controller for heat as an example as the desired temperature is reached, power input is reduced automatically (delaying the firing point on the triac) and enough energy is added to maintain temperature. In a cheap mechanical thermostat power is all on or all off producing overshoot and undershoot. This is unacceptable in many industrial applications.

For an application of this circuit see **Hatching Chicken Eggs with Arduino**

See **Controlling Low-Voltage Driveway Lights with the ATMEGA168/Arduino** In that circuit we use a CdS photocell instead of a thermistor, but the circuit is identical.

- **MOC30xx Opto-Coupler with Diac Output** PDF file
- **Hardware Interrupts Demo and Tutorial for Arduino**


```
/*
Purpose: to detect zero crossing pulse at INT0 digital pin 2,
which after delay switches on  a triac.
Power output to triac activated by external switch.
*/

#define triacPulse 5
#define SW 4
#define aconLed 12

int val;

void setup()  {
  pinMode(2, INPUT);
  digitalWrite(2, HIGH); // pull up
  pinMode(triacPulse, OUTPUT);
  pinMode(SW, INPUT);
  digitalWrite(SW, HIGH);
```

```
  pinMode(aconLed, OUTPUT);
  digitalWrite(aconLed, LOW);
 }

void loop() {
 // check for SW closed
 if (!digitalRead(SW))   {
  // enable power
  attachInterrupt(0, acon, FALLING);
  // HV indicator on
  digitalWrite(aconLed, HIGH);
 }  // end if
   else if (digitalRead(SW)) {
     detachInterrupt(0); // disable power
     // HV indicator off
     digitalWrite(aconLed, LOW);
   }  // else
 } // end loop



// begin ac int routine
// delay() will not work!
void acon()
  {
   delayMicroseconds((analogRead(0) * 7) + 200); // read AD1
   digitalWrite(triacPulse, HIGH);
   delayMicroseconds(200);
   // delay 200 uSec on output pulse to turn on triac
   digitalWrite(triacPulse, LOW);
  }
```

- In Depth Look at AC Power Control with Arduino
- YouTube Video for Arduino AC Power Control
- Four part series:
- Experimenting with the PCA9555 32-Bit GPIO Expander with Arduino
- PCA9555 32-Bit GPIO Expander with Arduino and a 4X4 Keypad
- PCA9555 32-Bit GPIO Expander with Arduino Using Interrupts
- PCA9555 32-Bit GPIO Expander with Arduino and LCD Display
- YouTube Video for Series

Added June 7, 2013:

- Rotary Encoder Used with Arduino Interrupts
- Connecting Arduino to a 74C164 Shift Register
- Using the Mc3479 Stepper Motor Controller with Arduino
- Build a Thermocouple Amplifier
- My YouTube Channel

You Tube Arduino Microcontroller Video Series March 2012:

- Introduction to Arduino
- Part 1: Programming Arduino Outputs
- Part 2: Programming Arduino Inputs
- Part 3: Programming Arduino Analog to Digital Conversion
- Part 4: Programming Arduino Pulse Width Modulation
- Arduino stepper motor control

- Quick navigation main page:
- Arduino Microcontroller Projects
- General Electronics Learning and Projects
- Raspberry Pi and Linux
- Connecting a PC Printer Port to Electronics with Python
- Microchip PIC 18F2550
- PICAXE Microcontroller

- Gen. Electronics
- My YouTube Channel
- Raspberry Pi & Linux
- Arduino Projects
- PIC18F2550 in C++
- PIC16F628A in Assembly
- PICAXE Projects

- Web Master
- Bristol VA/TN
- E-Mail
- Environmentalism
- US Constitution
- Religious Themes 1
- Religious Themes 2