In Depth Look at AC Power Control with Arduino

by Lewis Loflin

- For related material see:
- **Hardware Interrupts Demo and Tutorial for ATMEGA168/Arduino**
- **Basic Triacs and SCRs**
- **Solid State AC Relays with Triacs**

In the above video and the code below we take an in depth look at the hardware for using Arduino interrupts to control AC power through a triac. Using a zero-crossing detector Arduino will detect the pulse then calculate a delay to control the power output to a load. The complete circuit schematic is **here**.

For more on basic AC voltage see my video **Basic Electronic Power Supplies**

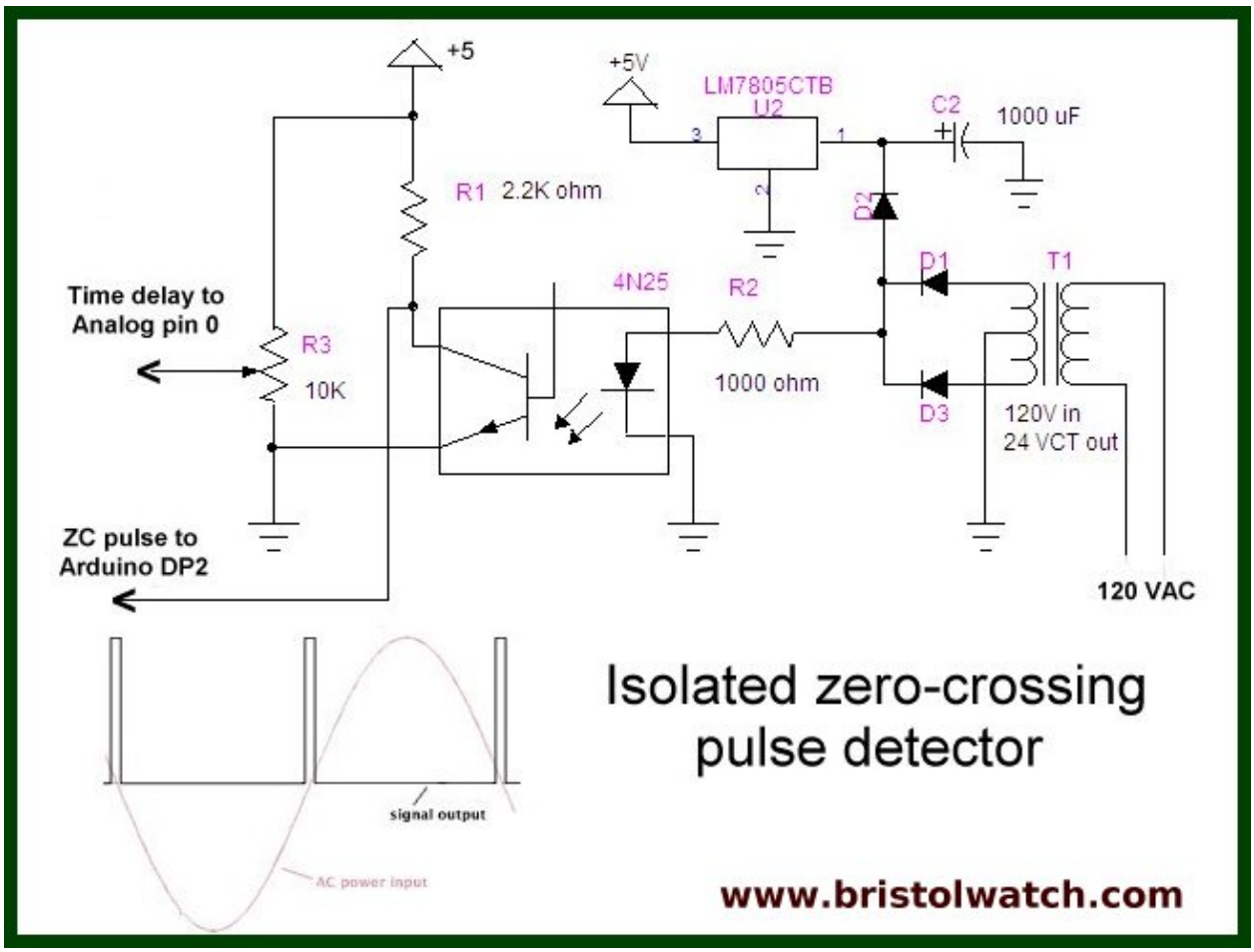

Fig. 1

Fig. 1 shows the 5-volt power supply for Arduino but includes blocking diode D2. On the cathode side we have filtered DC which is regulated to 5-volts through U2. On the Anode side we have unfiltered raw 120 Hz DC going to the LED in the 4N25 opto-coupler. The output from the photo-transistor collector goes to digital pin 2 of Arduino to interrupt 0. Potentiometer R3 goes analog pin 0 and is used to calculate the time delay for the triac firing pulses.

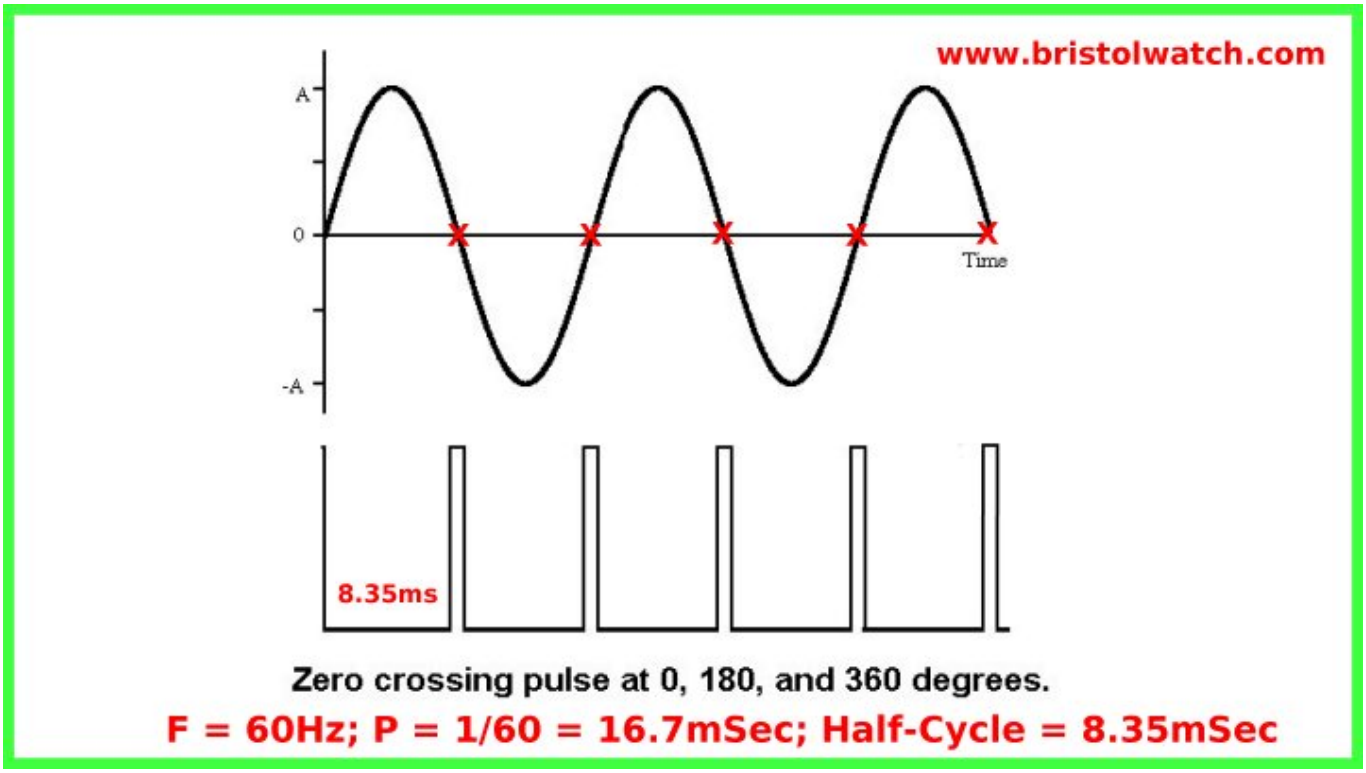

Fig. 2

Fig 2 shows the relationship of the zero-crossing pulse with the AC sine wave. By detecting the pulse and programming a delay one can control the power output level to a AC load.

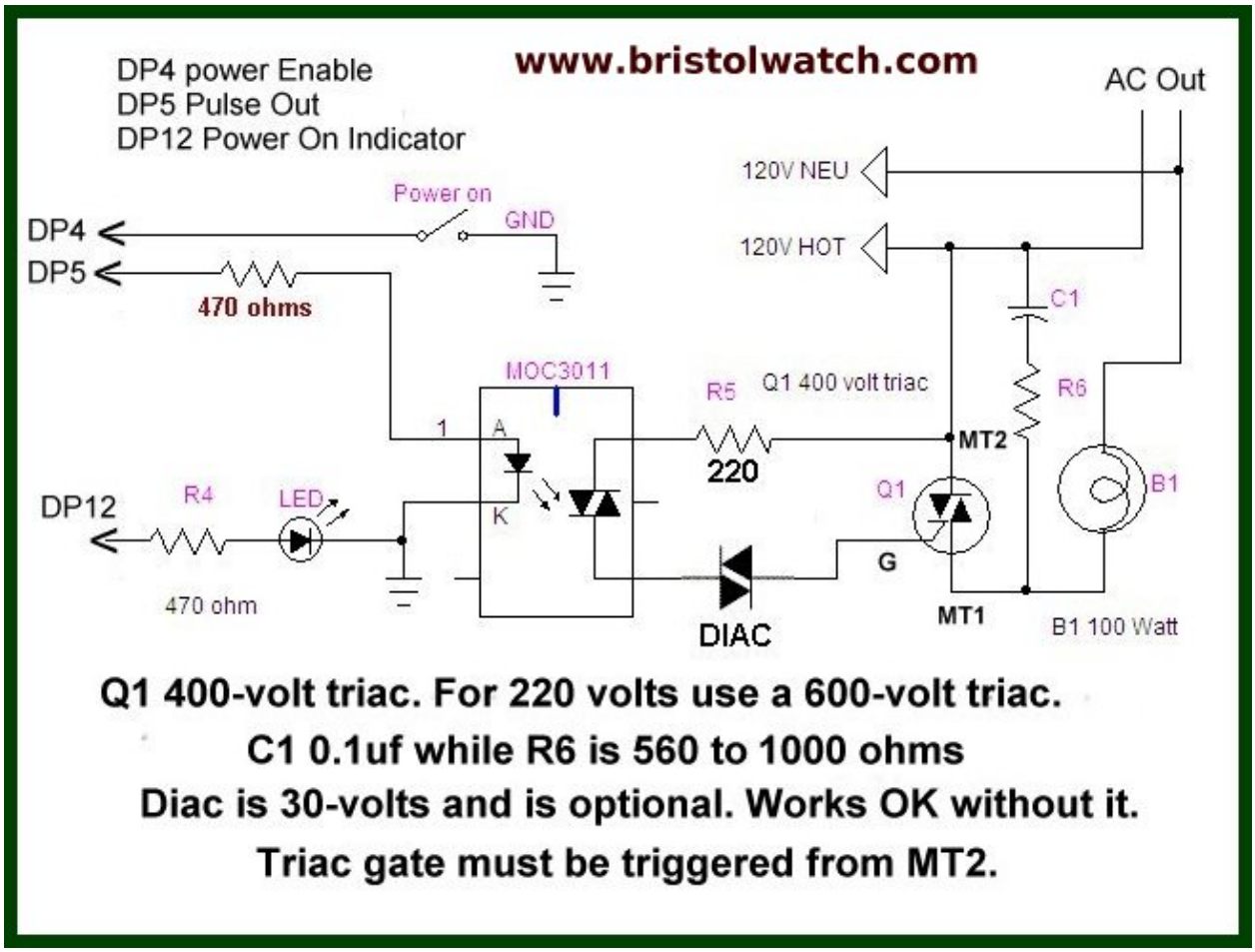

Fig. 3

Fig 3 shows the triac firing circuit. The MOC3011 opto-coupler uses a photo triac as opposed to a transistor. Pulses synchronized to the AC sinewave half-cycle are output from Arduino digital pin 5 to the LED in the MOC3011, which also serves to isolate the high voltage AC from the low-voltage components.

Pressing the power switch will enable trigger pulses to the MOC3011 while the LED on digital pin 12 is a power on indicator. C1 and R6 form a snubber circuit for inductive loads. Without a snubber switching noise from inductive loads will cause miss-firing of the triac.


```
/*
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 interrupt routine
// delay() will not work!
void acon()
{
  delayMicroseconds((analogRead(0) * 6) + 1000); // read AD1
  digitalWrite(triacPulse, HIGH);
  delayMicroseconds(200);
  // delay 200 uSec on output pulse to turn on triac
  digitalWrite(triacPulse, LOW);
}
```

- 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

- Testing the Keyes IR Sensor Module with Arduino
- How to Connect Easy Driver Micro-Stepper Controller to Arduino

- Connect Arduino to LCD Display with 74164 Shift Register
- Arduino with LCD Display and DS18B20 Temperature Sensor
- Below has differing code from the above. Works the same.
- Arduino with LCD Display and DHT11 Temperature-Humidity Sensor
- In Depth Look at AC Power Control with Arduino
- 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 Interfacing PCA9555 to Arduino

Stepper Motors

- Considerations for Using Stepper Motors
- How to Connect Easy Driver Micro-Stepper Controller to Arduino
- Using a Unipolar Stepper Motor with a Arduino
- Using the MC3479 Stepper Motor Controller with Arduino
- Connecting the Arduino to a L298N H-Bridge
- L298N Motor Controller Theory and Projects
- Arduino stepper motor control YouTube

- 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

website hits counter