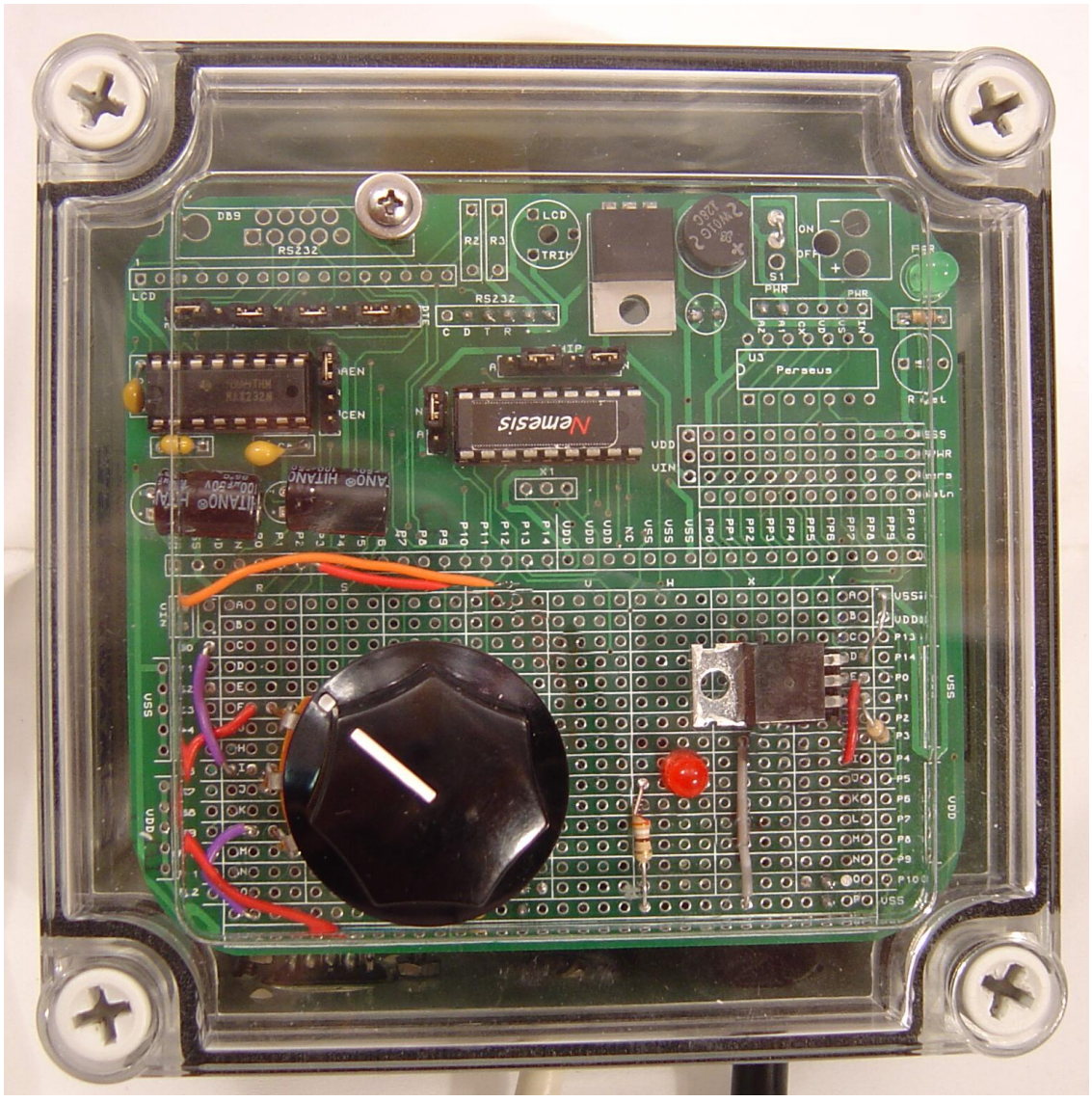
Build a Pool Timer
as seen in
September 2005 of Nuts & Volts Magazine

Pick up an issue at
http://www.nutsvolts.com



Kronos Robotics and Electronics
www.KronosRobotics.com

Last year about this time I did a floating pool light project. Judging from the responses I got, (and still get), it was a great success. So I think it's time for another pool project.

Before I get started, let me give you some background on how this timer evolved. Back when I created the floating light article I also created a pool timer.

Figure 1

Shown in Figure 1, this timer had a clock-like interface and various buttons for control. You set each hour you wanted the pump to turn on. The yellow LED in the middle was the AM/PM indicator, so you had 24hour control over the pump.

The timer worked ok, but was a bit too complicated for anyone else in the family to use. I promised my wife that next year I would build a simpler timer.

One factor surfaced as I used this timer as well as others: it was more a factor of duty cycle and less a factor of actual times as to when to turn on the pump. For instance, I found that if the pool needed extra cleaning because of a storm or heavy use, that I would set the timer for **ON 2 hours and OFF 1 hour**. Once the pool was cleaned to my satisfaction, I would drop the timer back to **ON 1 hour and OFF 2 hours**.
What I really needed was a timer control that would let me set a long term duty cycle for the pump over a 3 or 4 hour period.

Let's take a look at a few requirements for the new timer.

•Ability to control duty cycle
•Simple control system
•Self Contained (No AC adapters)
•Ability to override the timers
•Ability to reset the timers

My first prototype used two knobs. One knob was used to control the on-time, while the second knob was used to control the off-time. While this worked, it required you to fiddle with two knobs to override the timers. Also, I felt I could make the interface simpler.

What I ended up with was a single-knob control system. The only indicator was a red LED to indicate the status of the control relay. The actual position of the knob gave the only real feed back needed to control the timers.

The Control

The fact that the knob is the only control makes the timer very easy to set. By placing the knob indicator in the middle position, the pump is on two hours and off two hours. By moving the knob in ether direction, you can change the amount of on and off time in proportion to the position of the knob.

Figure 2

Figure 2 shows how the various positions affect the duty cycle. Placing the knob all the way to one side or the other lets you override the timers completely. This way you can turn the timer off or on at will.

Note that the timer is based on a four hour cycle. This can be changed in the program code to whatever you wish. I will explain how to do this in more detail later.

Construction

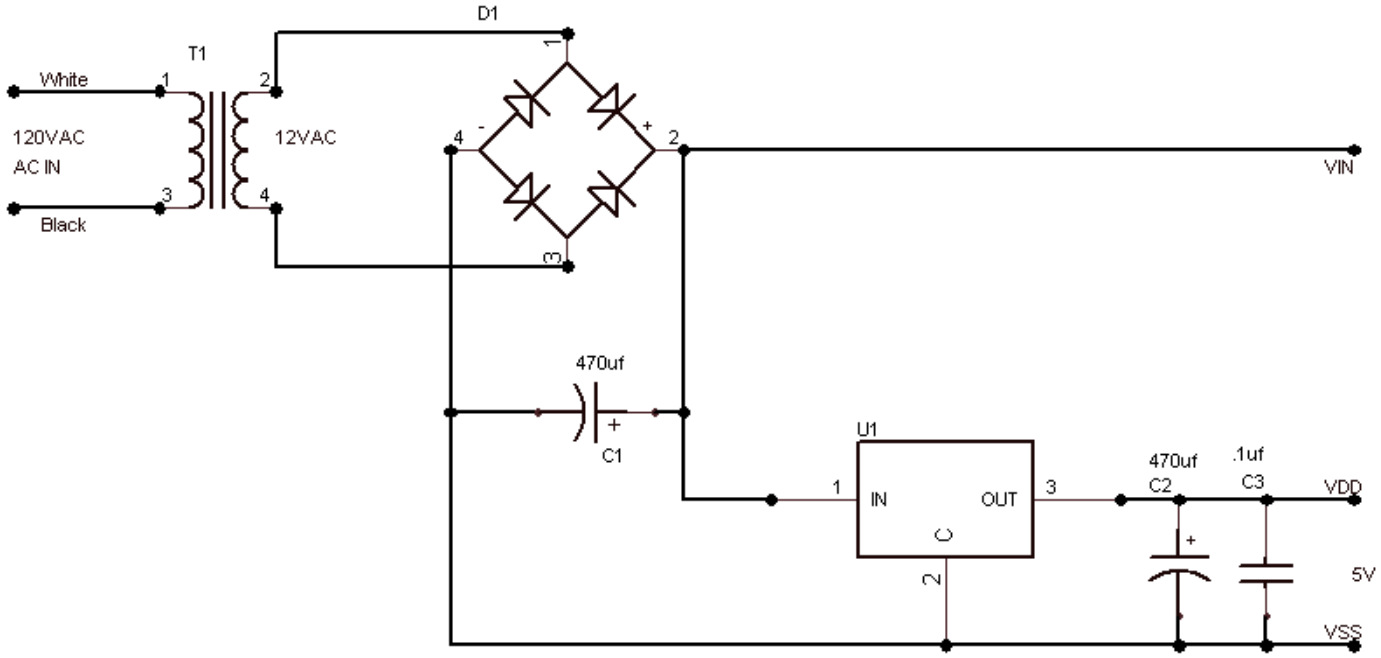
Let's take a look at the circuit. The fact that we only have one control keeps the parts-count very low. Most of the components are actually used in the power system.

In order to keep the construction details simple, I will look at each sub system individually so you can make you own decisions based on the components you have in your junk box.

Power System

Schematic 1 shows the power system. I used a very small low power 300ma transformer to supply both the power to the microcontroller circuit, as well as to power the controller relay. It's important that you use a 12v transformer so that you can power the controller relay from a tap off the rectifier bridge. If the relay you decide to use draws more than 250ma, you will need to use a larger transformer.

U1 is a simple 1Amp 7805 regulator. You can use just about any 5v regulator to power the logic circuit. The 2, 470uf capacitors are very important, and if you have any thing larger in your junk box, you can use those.



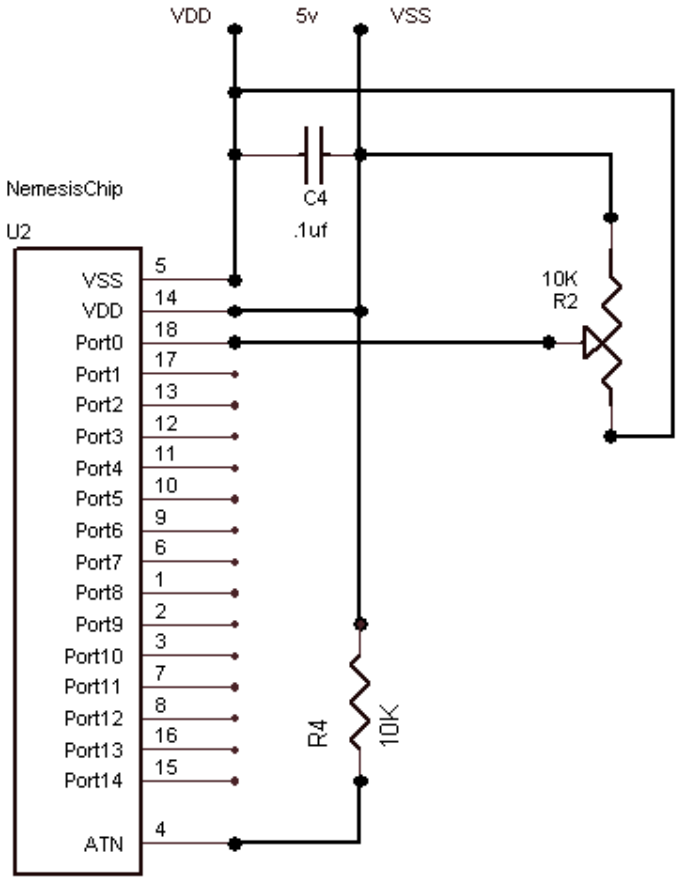
Schematic 1

Logic System

I used the Nemesis microcontroller because its small, cheap, and you don't need an expensive programmer to program it. The Nemesis has 7 AtoD ports that we can use to read the position of the knob, and plenty of ports for driving the indicator and control system. Other than that, you only need a single .1uf and 10k resistor for support components.

Schematic 2 shows how we connected R2: the control knob. Any potentiometer from 1K to 100K can be used. Just make sure it is not an audio taper.

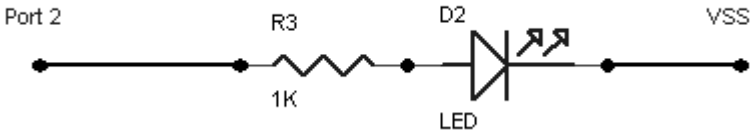
In order to program the Nemesis, you will also need a RS232 interface. You can build one with a MAX232 chip or purchase one for less than $10 from the Kronos Robotics web site. The software for programming the Nemesis is free and can be downloaded from the web site as well.



Schematic 2

Indicator System

Other than the knob itself, you only need a single LED to display the state of the control relay.

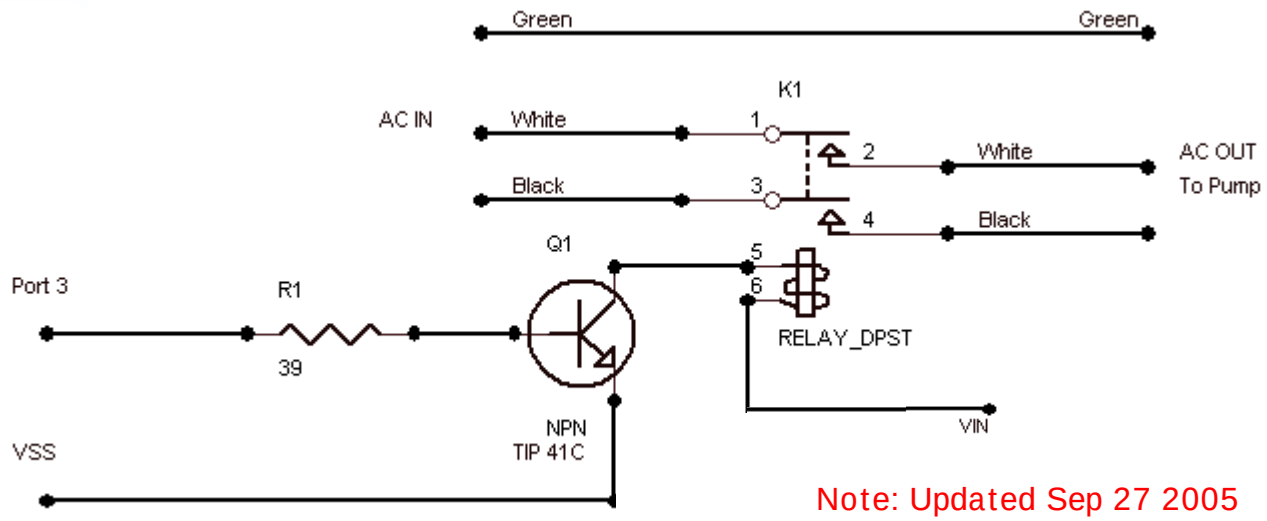


Schematic 3

Any LED will work. You will also need a resistor, used to drop the 5v down to an acceptable level, as shown in schematic 3.

Control System

The control system consists of a TIP41 Darlington transistor and a DPDT 12v relay.



Schematic 4

I like using the TIP 41 in situations like this, as it has a bypass diode built-in to keep the relay from destroying the transistor when the field collapses.

The relay can be any 12v DC relay, as long as the contacts can handle the pump you are controlling. I am currently using a 12v relay with 20Amp Contacts. Originally, I used a 30Amp 12v automotive relay. These relays will work, but will draw a bit more current than your typical DPDT 12v relays. If you do decide to use an automotive relay, make sure you switch the black (or hot) power lead. If your pump is 220v, I really recommend a DPDT relay so you can switch both sides of the circuit.

When I build circuits like these, I like to purchase a 50' or 100' extension cord and cut it down to a couple feet. This gives me a nice male and female connector as well as any length of cord I may need. This tends to be cheaper than purchasing the special plugs needed. If you are running 220v you may just need to hardwire the circuit anyway.

PCB

Kronos Robotics has several PCBs for the various microcontrollers. The Athena WorkBoard PCB is perfect for this project. It has provisions for the connecting AC with the rectifier and regulator on board. You can also use the board to program the chip as there are provisions for an RS232 driver. There is also a very generous prototyping area on the PCB for adding the potentiometer, indicator, and transistor.

Note that the PCB is also available in a Carrier board configuration which makes it extremely easy to prototype your circuit.

Case

I purchased Industrial Control Enclosure from All Electronics to house my project; however, these are no longer available. I recommend you choose your PCB and other components first. You can then determine the size of the case needed to enclose your circuit, transformer, and control relay.

My pool shed tends to leak, so I want to make sure the case was water proof. I sealed all holes with hot glue and made sure all cables exited the case at the bottom or underside of the case.

Software

In order for the Nemesis to do its job we need to load it with a small program. I say small because this program only uses about 1/20th of the available memory.

The Program

```
Nemesis
'Pool Timer

  '_____-
  'Some Variables
  '_____-
  dim onval,offval,aval
  dim state,maincounter,littlecounter
  clearall

  '_____-
  'A little Setup
  '_____-
  atodinit 0
  output 2
  output 3
  low 2
  low 3
```

```
'_____________________________________________
'Main Loop
'_____________________________________________
loop:

  '_____________________________________-
  'Get the Knob Value and save its settings
  '_____________________________________-
  atod 0,aval
  onval = 255 - aval
  offval = aval

  print maincounter," ",littlecounter," ",onval," ",offval," ",state

  '_____________________________________________-
  'Counter Section
  '_____________________________________________-
    littlecounter = littlecounter + 1
    if littlecounter > 60 then
      maincounter = maincounter + 1
      littlecounter = 0
    endif
    longpause 4,250 '1 Second


  '_____________________________________________-
  'Check Knob Status Section
  '_____________________________________________-
  if state = 0 then
   if maincounter >= offval then
    state = 1
    maincounter = 0
    goto loop
   else
    low 2
    low 3
    goto loop
   endif

  else
  if maincounter >= onval then
    state = 0
    maincounter = 0
    goto loop
  else
    high 2
    high 3
    goto loop
  endif

  endif

  goto loop
```

The knob is read once each second and its values are checked against internal timers. In order to get the length of time needed, I use two counters. They are called littlecounter and maincounter. Think of littlecounter as an inner counter. It is incremented once each second, and when it reaches 60, the maincounter is incremented.

By doing this, the maincounter represents minutes. You can shorten or lengthen this interval by changing the littletimer threshold.

Final Thoughts

A word about safety. If you decide to build this circuit, I can't place enough emphasis on how important it is to use a CFGI outlet. If you are not currently using a CFGI outlet then I recommend you install one.

The Nemesis is a very powerful microcontroller and supports a high level basic language as well as inline assembly. We haven't even touched on the kinds of enhancements you could add to this timer. The Nemesis also has a command for controlling a TW523 X10 transceiver. By utilizing this device you could use an appliance module to control the pump.

Be sure to visit the Kronos Robotics web site for updates to this and other articles.

Sources

The Kronos Robotics web site is located at: www.kronosrobotics.com

Nemesis Microcontroller	http://kronosrobotics.com/xcart/customer/product.php?productid=16406
Athena WorkBoard PCB	http://kronosrobotics.com/xcart/customer/product.php?productid=16460
Athena WorkBoard Deluxe	http://kronosrobotics.com/xcart/customer/product.php?productid=16457
Athena WorkBoard Basic	http://kronosrobotics.com/xcart/customer/product.php?productid=16473
EZ232 Driver	http://kronosrobotics.com/xcart/customer/product.php?productid=16167
7805 5v Regulator	http://kronosrobotics.com/xcart/customer/product.php?productid=16208
Bridge Rectifier	http://kronosrobotics.com/xcart/customer/product.php?productid=16136
Potentiometer	See Text
Transformer	Radio Shack #273-1385B
Relay	See Text
TIP41	Jameco #33081
Athena Compiler	Free from the Kronos Robotics Web Site