Do-it-yourself, Perl-controlled irrigation system

DON'T BLAME THE GARDENER!

In this month's Perl column, we will introduce a system to water your plants while you are away from home. With a little help from Perl, a friendly, Linux-based irrigation system waters your plants twice a day.

BY MICHAEL SCHILLI

Glenn Jenkinson, Fotolia

THE AUTHOR

Michael Schilli works as a Software Developer at Yahoo!, Sunnyvale, California. He wrote "Perl Power" for Addison-Wesley and can be contacted at *mschilli@perlmeister.com*. His homepage is at *http://perlmeister.com*.

During the summer months, if you take just a couple days of vacation and forget to ask your neighbor to water your plants while you are away, the plants on your balcony might not survive the drought. Instead of heading down to the supermarket to replace the dried-up victims when you get home, you might prefer to leave the task of watering your plants to a simple, do-it-yourself pump (Figure 1) [1].

Cutting Your Losses

If you rely on software to control a water supply, there's always the danger of flooding your apartment if an unforeseen bug strikes home. However, there is an easy way to avoid this – just use a bottle as your water supply to keep the volume down to a level that will not cause you any headaches [2].

My wife doesn't like the sight of glued-up water bottles in our apartment, so I went for a more attractive solution that features a doggy-biscuit holder with a folding lid, which I bought for US$ 4.75 down at the local hardware store, and an aquarium pump (US$ 6.75, plus postage and handling) that I found on Ebay.

Figure 1: A 1001 uses for a dog-biscuit holder. The integrated pump sends water through the hose and out the window to my balcony.

Figure 2: Don't forget to top up the water tank from time to time.

For the next step, I used my trusty drill, which was fitted with a grinding attachment, to grind a slot at the top back of my impromptu water tank. I then fed the plastic hose (purchased from the local do-it-yourself store) and the power lead for the pump through the slot with the lid closed.

The hose then runs through the window to a position above the plant pots out on the balcony.

Above every plant that needs watering, there is a tiny hole in the hose just big enough to let water drip down into the pot. Make sure the holes you make aren't too big. If they are, use glue to make them smaller or fill them in.

I used holt-melt adhesive to make sure the aquarium pump stays firmly stuck to

the bottom of my water tank. To top up the tank, you just flip up the red lid and pour in the water (Figure 2).

Watering Command via Line Power

An X10 module [3] switches the pump on and off. You can use the X10 protocol to send signals via line power circuits [4]. In this case, the transmitter is connected to the Linux machine's serial (or

Listing 1: water

```
01 #!/usr/bin/perl
02 use warnings;
03 use strict;
04
05 use Device::SerialPort;
06 use ControlX10::CM11;
07 use Log::Log4perl qw(:easy);
08 use Waterscore
09   qw(waterscore);
10
11 my $HOUSE_CODE = "K";
12 my $UNIT_CODE  = "11";
13 my $SERIAL   = "/dev/ttyS0";
14 my $BAUDRATE = 4800;
15 my $LOCATION = "USCA0987";
16
17 die "You must be root"
18   if $> != 0;
19
20 Log::Log4perl->easy_init(
21 {
22   level => $DEBUG,
23   file =>
24     ">>/var/log/water.log"
25 }
26 );
27
28 my $score =
29   waterscore($LOCATION);
30
31 if ($score < 20) {
32   INFO "No water.";
33   exit 0;
34 }
35
36 my $serial =
37   Device::SerialPort->new(
38   $SERIAL, undef);
39 $serial->baudrate($BAUDRATE);
40
41 # Address unit
42 ControlX10::CM11::send(
43   $serial,
44   $HOUSE_CODE . $UNIT_CODE);
45
46 # Turn water pump on
47 INFO "Water on";
48 ControlX10::CM11::send(
49   $serial, $HOUSE_CODE . "J");
50
51 sleep($score / 10);
52
53 # Turn water pump off
54 INFO "Water off";
55 ControlX10::CM11::send(
56   $serial, $HOUSE_CODE . "K");
```

Figure 3: The X10 receiver, with the aquarium pump's power cable plugged in.

USB) port and uses an X10 PC interface to feed signals into the line circuit. I used Model HD11A, which is available at the x10.com website or on Ebay.

The receiver is connected to another power outlet on the same circuit and switches the aquarium pump attached to it on and off when it receives the appropriate signals from the sender.

Make sure to use a X10 "appliance module" for this; the "lamp module" won't work.

Controlling the X10 sender's serial port with a Linux box is easy. You need to run the *water* script (see Listing 1) as *root* to make sure you have access to your computer's first serial port, */dev/ttyS0*, and the X10 transmitter must be connected to this port.

You can use your USB port to control another X10 transmitter model if you like. The Wish project [5] has some free software for this task.

The values for the house code (*K*) and the unit code (*11*) in the script are required to address the receiver, which is also set to *K* and *11*, as you can see in Figure 3. These values have to be unique on your line circuit.

The *water* script uses the *Device::SerialPort* and *ControlX10::CM11* modules from CPAN for the tricky stuff with the serial port and to handle the intricacies of the X10 protocol.

In X10, the command for switching on the receiver is *J*, and *K* switches the re-

Figure 4: Weather codes for various places called "London".

ceiver off again. Before you send a command, you first need to address the receiver. Furthermore, each command must be preceded by the house code.

Green Thumbs

Plants need more water in hot weather, and none at all if it rains. To find out what the weather is doing, the *water* script queries the weather service at *weather.com* on the Internet for the current temperature, humidity, and general weather forecast.
The *water* script then uses these values to generate an irrigation score between 0 and 100.

Zero means the plants don't need watering, and 100 means they need plenty of water.

Listing 2: Waterscore.pm

```
01 #############################
02 package Waterscore;
03 #############################
04
05 use base Exporter;
06 @EXPORT_OK = qw(waterscore);
07
08 use strict;
09 use warnings;
10 use Weather::Com;
11 use Log::Log4perl qw(:easy);
12
13 my $PARTNER = "1031443804";
14 my $LICENSE =
15      "8d88149d90fec933";
16
17 #############################
18 sub waterscore {
19 #############################
20   my ($loc) = @_;
21
22   my $w = Weather::Com->new(
23     current    => 1,
24     units      => "m",
25     timeout    => 250,
26     partner_id => $PARTNER,
27     license    => $LICENSE,
28   );
29
30   my $now =
31     $w->get_weather($loc);
32   return undef
33     unless defined $now;
34
35   my ($cond, $temp, $humi) =
36     map { $now->{cc}->{$_} }
37     qw(t tmp hmid);
38
39   DEBUG
40     "$cond, $temp C, $humi%";
41
42   my $score = 50;
43   $score += 50 if $humi < 50;
44   $score -= 30
45     if $cond =~ /Showers/;
46   $score -= 80
47     if $cond =~ /Rain/;
48   $score += 80
49     if $cond =~ /Sunny/;
50
51   if ($temp > 20) {
52     $score +=
53       ($temp - 15) * 10;
54   }
55
56   $score = 100
57     if $score > 100;
58   $score = 0 if $score < 0;
59   DEBUG "Score: $score";
60   return $score;
61 }
62
63 1;
```

Figure 5: Water is sprayed through a hole in the hose onto the plants on our balcony.

If the score is below 20, *water* doesn't water, whereas the irrigation period grows by one second with each block of 10 points added to the score.

The pump will run for a maximum of 10 seconds, but you can adjust the time to reflect your pump capacity and the water requirements of your plants.

The irrigation score is calculated by another module called *Waterscore.pm* (see Listing 2). The calling script, *water*, asks *Waterscore.pm* to export the *waterscore* function into the script's namespace. Perl's *Exporter* module handles this task.

Waterscore.pm contacts the weather service, *weather.com*, passing in a partner ID and a license key, both of which are available for free after you register with a valid email address.

The CPAN *Weather::Com* module nicely abstracts getting the data from the web service, and the module converts the XML response to a developer-friendly Perl structure.

Weather.com provides a weather service for regions with codes such as "USCA0987" (for San Francisco). The *location* script helps to locate the code for your local weather region.

Figure 4 shows the *location* script output with an argument of "london" (Listing 3). Besides a number of towns called "London" in the US, this also returns an entry for London, UK: UKXX0085.

Waterscore.pm uses a couple of tricks, devised by empirical methods, in order to calculate the score.

If the weather report contains the words "*Rain*" or "*Showers*", it deducts 80 points, to make sure that the plants will not be watered, unless it happens to be very hot.
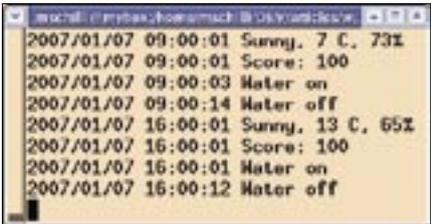
Low humidity increases the score – and thus the probability that the plants will be watered – along with the irrigation period.

If the report contains the word "*Sunny*" in the {*cc*}->{*t*} hash entry, and if temperatures of above 15 degrees Centigrade are reported, this will also affect the score.

If the *Weather::Com* constructor is called with a true value for the *current* parameter, the method called immediately after this, *get_weather()*, will get the current weather values and not the forecast we're not interested in. The *units* parameter is set to "*m*" (for metric) to get the temperature in Centigrade rather than Fahrenheit.

You need to install the *Waterscore.pm* module in a path where *water* will be able to locate it (in /usr/lib/perl5/site_ perl, for example). A cronjob running under the *root* account, *00 9,16 * * * /usr/bin/water*, runs the irrigation script once in the morning and once in the evening to make sure the plants get a good supply of water.

You can check the logfile in */var/log/ water.log* to see how much water your plants have received and when they received it.

Make sure the hose stays at about the same height as the water tank. Cheap aquarium pumps are leaky when they're turned off and the law of gravity will slowly drain the reservoir if only one hole in the hose gets positioned significantly lower than the water tank.

For advanced watering equipment, check out **dripdepot.com**, where all kinds of irrigators and watering supplies can be purchased for little money.

Just make sure you top up your water tank from time to time, and enjoy your summer vacation! ∎

Listing 3: location

```
01 #!/usr/bin/perl -w
02 use strict;
03 use Weather::Com;
04
05 my $w = Weather::Com->new();
06
07 $ARGV[0]
08   or die "usage: $0 city";
09
10 my $locs =
11   $w->search($ARGV[0]);
12
13 for my $loc (keys %$locs) {
14  printf "%-20s %s\n",
15    $locs->{$loc}, $loc;
16 }
```


Figure 6: The logfile tells you how much and when your plants were watered.

INFO

[1] Listings for this article: *http://www.linux-magazine.com/ Magazine/Downloads/77/Perl*

[2] *Make*: *http://makezine.com*, Volume 08, page 22.

[3] X10: *http://en.wikipedia.org/wiki/X10_ %28industry_standard%29*

[4] Michael Schilli: "*Inside*", *http://www. linux-magazine.com/issue/50/Perl_ Building_a_Jabber_Bot.pdf*

[5] Wish: *http://wish.sourceforge.net*