How I Made

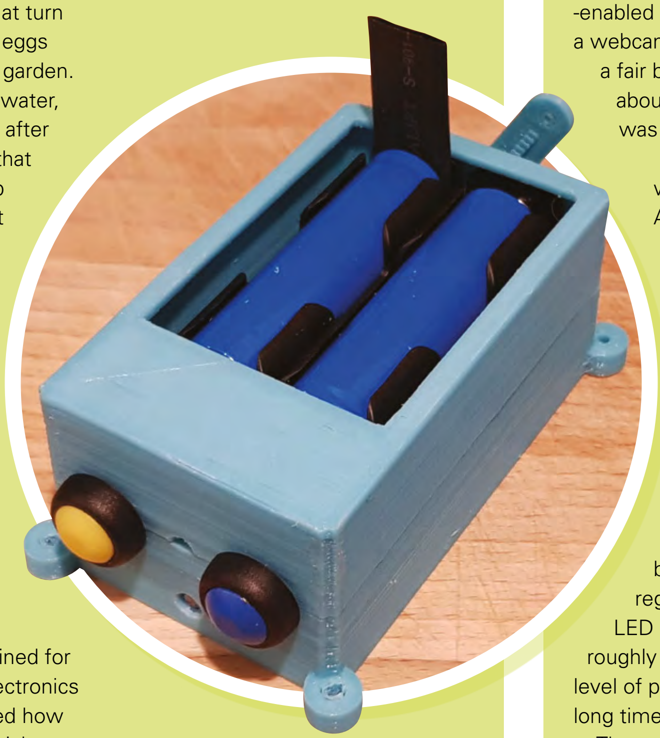
A CHICKEN COOP OPENER

The making of some fowl technology

By **Andrew Lewis**

Chickens are great. They're like little dinosaurs that turn slugs and plants into eggs and fertilizer for your garden. Aside from food and water, chickens mostly look after themselves. The only real chore is that they need to be shut into their coop when the sun sets, and let back out again when the sun rises. If you find yourself away from home unexpectedly, you inevitably end up worrying about the chicken door. After a month of chicken door anxiety, I decided that I'd find a technological solution to the problem.

The requirements for a door opener are very straightforward. It needs to open a wooden door when it's light outside, and close it again when it's dark. It also needs to lock the door closed if we want to keep our chickens confined for some reason. I'm no stranger to electronics or making things, but I was surprised how much of a challenge this 'simple' chicken door opener presented. The chicken door mechanism isn't complicated, but in our case, running a power cable to the chicken run wasn't practical. That meant the door opener needed to run from batteries, and my initial plans for a fancy Internet of Things -enabled chicken door with a phone app and a webcam were modified into something a fair bit simpler. I had to think very hard about how much power the project was going to need.



Above
The MK2 chicken door opener assembled in a 3D printed case, with batteries

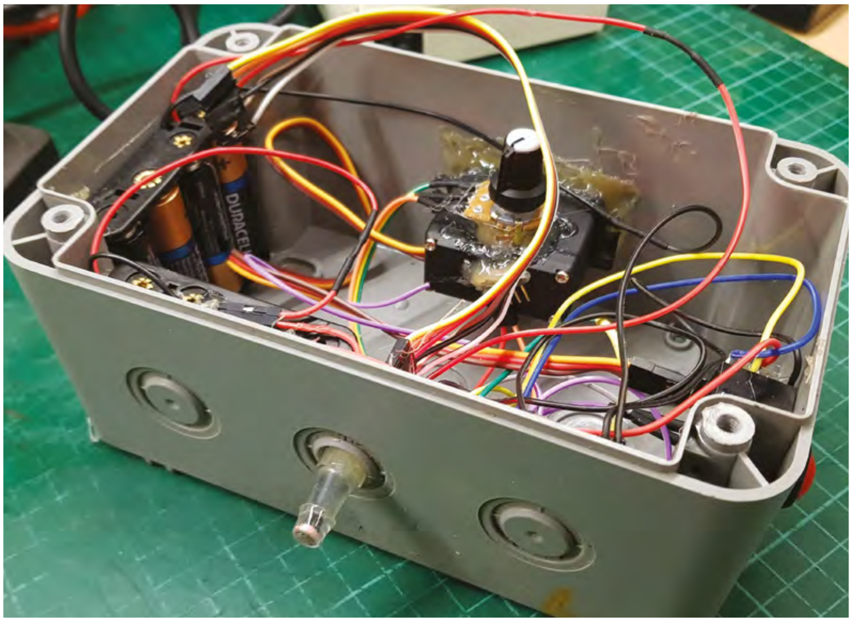
The first issue I had to consider was idle power consumption. A standard Arduino Uno uses roughly 40–50 mA of power at idle. Even if you put the Arduino to sleep when it's not being used, the on-board monolithic voltage regulator still uses a fair amount of power. The 5 V version of the Arduino Pro Mini is much more frugal, using around 20 mA when idle, and 3 mA when asleep. A little bit of internet research told me that bypassing the on-board power regulator and disabling the power LED could reduce this number to roughly 0.005 mA during sleep mode. That level of power could be sustained for quite a long time by a bank of four AA batteries.

The type and power consumption of the motor was also a factor in the design. I initially wanted to use a geared motor to control the door, but realised that there would be a couple of problems with this. The geared motor would lock into position when the power was disconnected, which would mean that I couldn't move the door by hand if the mechanism failed. I would also need to add sensors or limit switches to monitor the position of the door. I already had some high torque servos in my workshop, and I decided to use one of those. The current draw from a permanently activated servo would have drained any batteries I had in a few hours, so I added a relay that would only power the servo when the door needed to move.

Although the power used by the light-dependent resistor is trivial, I decided to apply the same principle that I had used with the servo and only supplied power when a reading was being taken. In addition to the LDR, I added a potentiometer to set the threshold light level that the door would trigger at, and a simple push-button that would be used to close and lock the door for ten minutes. I thought that ten minutes would be enough time for us to clean or move the coop, and I didn't want to accidentally lock the chickens in for an extended period. I came to regret this decision later, as we were chasing three fugitive birds around the vegetable garden.

I assembled the door opener into a waterproof electrical box, programmed the Arduino, and gave the system a test. It worked initially, but started to misbehave when any load was put onto the servo. I realised that the current being drawn from the batteries was too great, and the servos were making the Arduino brown-out and reset. I modified my design, added a separate battery bank just for the servo, and tried again. This time the door operated as expected, so I sealed everything into the box and fitted the unit to the chicken coop.

Several weeks later, the door of the coop failed to open. This failure was sooner →



Above
The MK1 chicken door opener, with lid removed to show the potentiometer and battery boxes. It's a mess of wires, and that makes it easy to knock something loose when the batteries are being changed

Below
The MK1 chicken door, installed on the coop. It's a bulky unit and it's difficult to set the light level using the potentiometer, but it does work

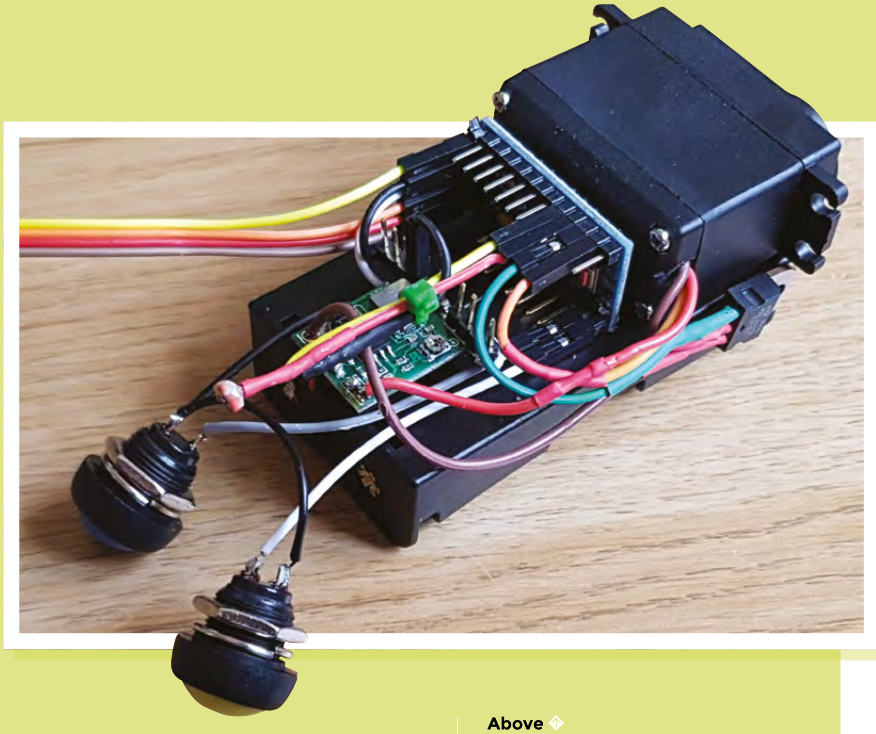
The Rocket Scream Low-Power Library

An Arduino Pro Mini is usually based on the Atmega328p processor, running at either 8 MHz or 16 MHz, depending on the version of the board. The Low-Power library by Rocket Scream (**rocketscream.com**) allows you to put the Atmega328p into a power-saving state using simple commands, which vastly improve longevity while running on battery power.

As an example, to make the chicken door go into power-saving mode every two seconds, I imported the Low-Power library and used the following command:

```
LowPower.powerDown(SLEEP_2S, ADC_
OFF, BOD_OFF);
```

This command tells the Arduino to sleep for two seconds, and to turn off the ADC (analogue to digital converter) and the BOD (brownout detector) during this sleep period. You can also do more advanced things with the library, like using hardware interrupts to wake the Arduino from sleep.



Above The MK2 chicken door opener with the all of the sensors and buttons connected to the Arduino. The PCB above the Arduino Pro Mini is a voltage regulator

than I had expected, but I replaced the batteries and ran another test. The door opened and closed properly, and I thought nothing more of it. The following morning, I faced the same situation again. I concluded that something must have happened to either the wiring or the Arduino, and chose to make a new improved version. I had a functional workshop, and so I thought that I would be able to make a better job the second time around.

I opted to use 18650 batteries on the new door opener. 18650 batteries have a much higher drain current and voltage than AA batteries, and are rechargeable. Two 18650 batteries should be enough to power the Arduino and servo for several weeks at a time, and the more compact batteries meant that I could reduce the size of the whole project significantly. Connected in series, the 18650 batteries give 7.4 volts when fully charged, which is serendipitously the same as the maximum recommended voltage for the servo.

CHICKEN DOOR MK2

I started the new build with another Arduino Pro Mini. I bypassed the internal voltage regulator and added a more efficient external buck (step-down) converter. I added a light-dependent resistor to an analogue pin, using a standard potential divider set up between a digital pin and the ground pin, so that I could power down the LDR when it wasn't being used. I added two buttons to the Arduino, and activated the internal pull-up on the pins I connected them to. The first button closed the door, just as it had on the first version of the chicken door opener. Unlike the original unit, the new door would stay closed until the second button was pressed to unlock it. Once the door had been unlocked by the second button, it would operate normally and close by itself when it got dark enough.

On the original chicken door opener, I used a potentiometer to set the trigger level for the door. I found later that it was very difficult to set the level this way, because it was easy to knock the potentiometer or move it accidentally. This time around I took advantage of the extra button I'd added, so that pushing both buttons together for three seconds would take a reading from the light sensor and use this as the threshold level to open and close the doors.

I used a TIP120 to control the power to the servo, rather than a 5V relay as I'd done in the original version. I did this mainly because I had one handy on my desk, but also because the TIP120 is smaller than a 5V relay and has no moving parts. I know that the TIP120 isn't very efficient by modern standards, but in this project it's only powered up for a few seconds at a time, and it's so over-powered for the job that it'll never dump enough heat to get even slightly warm.

I assembled all of the parts using the plastic battery holders as the main chassis, and fixed each of the components in place using double-sided neoprene tape.

I started the new build with another Arduino Pro Mini. I bypassed the internal voltage regulator and added a more efficient external buck converter

Above The MK2 chicken door, installed on the coop. The battery cover has been removed to show the 18650 batteries inside. This unit is more compact and much easier to set up

I 3D-printed a suitable case, and sealed the seams of the unit together using a hot soldering iron. I was feeling quite pleased with the project at this point, and it's a pity that it only worked for three days.

POINTS OF FAILURE

I'd made a couple of mistakes in my design, and misunderstood the cause of the original unit's failure. Firstly, I hadn't included a backwash diode, to absorb reverse current feeding back from the servo. This wasn't a big problem in the original version, because it used a relay and had an independent battery bank. In the new version of the door opener, the relay was replaced with a transistor, and wired to the same supply as the Arduino. This caused some stability problems.

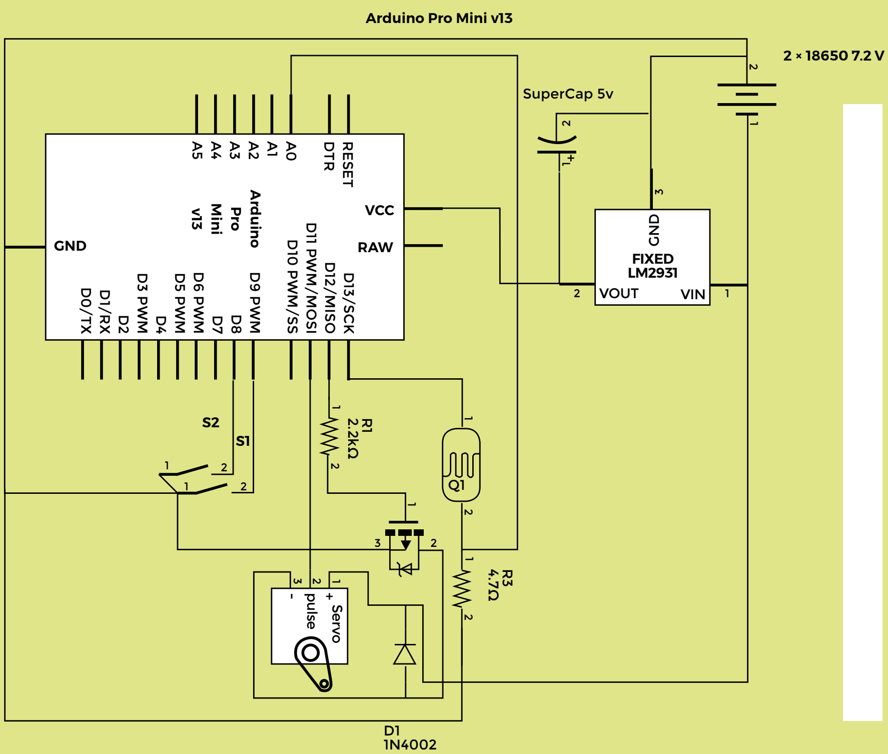
The most important issue was that the servo still seemed to be drawing too much current from the batteries. This confused me at first because I knew that the output current of an 18650 battery should be able to power the servo. Even more confusingly, after I'd tried to open and close the door a few times, the opener started working properly. It took me quite a while to realise that the problem wasn't inside the box, it was actually the environment that I'd put it in. Like most batteries, 18650 cells are very sensitive to cold conditions. The capacity rating printed on batteries is usually calculated for a temperature of around 25 degrees Celsius. The chicken door opener had worked perfectly through the warm summer months, but the capacity of the batteries had fallen as the temperature approached freezing. The final point of failure was that I'd used some unprotected batteries from my spare parts bin, and the voltage had dropped too low for them to be recharged.

IMPROVEMENTS

Once I realised the problem, the solution was simple enough. I added a 5V ultra-capacitor after the power regulator, and this made sure that the Arduino still receives enough current when the servo is draining all of the power from the battery. Capacitors are less affected by low temperatures than batteries are and, with a new set of protected 18650 batteries, the Chicken Door Opener MK2 has been working for several weeks without incident.

When the weather improves, I'll probably update the code on the Arduino so that the light level that triggers the door is stored in EEPROM rather than RAM. There's no rush to do this, because the ultra-cap can keep the Arduino powered up for several hours even without batteries installed.

Above The customers inspect our handiwork



Below The final circuit has proved reliable and easy to use