

**Dew Destroyer**

By Paul Surowiec

So, I had a couple of dew straps and would just hook them up to my 12V battery and then unplug them when I felt they had warmed up the corrector plate and eyepiece enough to prevent some dew. Then when things started to dew up again I would plug them back in. Not a very efficient or graceful process but it kind of worked.

I looked at both purchasing and building a dew controller to automate the procedure. Most dew controllers just reduce the power to the dew heater straps by one of two methods. One method is by lowering the output voltage. The other is by driving them by switching the power to them on and off and reducing the amount of time the voltage is turned on. This is controlling the duty cycle or pulse width modulation. One top of the line dew controller the **Kendrick DigiFire 10** includes one ambient temperature sensor and one optic sensor to control one of the power ports.

There had to be a better way to do this and there was. Low and behold one day on the Cloudy Nights Astronomy forum I read an intriguing post by a gentleman who goes by “mitaccio” from Hawaii. He had taken an Arduino micro controller project board, a temperature/humidity sensor and two external temperature sensors to control his two dew heater straps. He also uses a red LCD 2 line x 16 character display to display basic information. The dew controller keeps each of the two dew straps 10°F above the dew point. He also displays the outside temperature, humidity, dew point and temperature of each of the dew straps. In addition he use the ^ carrot next to the temp to indicate when the power was turned to each of the dew straps. I liked what I saw, but saw some shortcomings to what he had created and room for expansion and enhancements.

The Arduino is programmed in a C type of language on a PC or MAC, compiled and then downloaded to the Arduino via a USB connection. The IDE (Interactive Development Environment) to create programs is available for Windows, MAC and linux. I am not a C or C++ programmer, but am familiar with other Object Oriented Programming (OOP) languages. So I had a rough idea of what mitaccio was doing. I ordered the parts I needed, while waiting for them to arrive I downloaded the software and tried to compile it. Failure number one was instantly encountered, his program would not compile and I had tons of errors. After trading PM’s (Personal Messages) with mitaccio, I soon learned he had gotten help from a number of people on the internet and he didn’t have good understanding of his program. Biggest problem I found is that for the DHT11 Temperature/Humidity sensor and the two Dallas One-Wire Temperature sensors there are a myriad of different versions of driver libraries out there. On the good side, this forced me to learn and understand the C type of programming language on the Arduino. I did eventually find the correct versions of the libraries for the sensors and after about a week got the program to compile. So by that time my parts arrived and I was able to breadboard them and get it working spread out on my desk in a couple of days.

So now it came to expand, enhance and fix its shortcomings:

* Protect the system from having the power hooked up backwards by accident.
* Added a fused cigarette lighter plug to prevent shorts from damaging it.
* I expanded it to use 4 Dallas one wire sensors and control 4 individual ports.
* Added a separate power input for the dew strap heater circuits.
* Added heat sinks to 2 of the output driver transistors and vented the case so they can run at a full 5 Amps each. The other 2 are capable of at least 2 Amps each. More than enough for the largest of any commercially available dew straps.
* I added the ability to capture and store the high and low values of each sensor to the EEPROM.
* I added the ability to display and capture values in either Fahrenheit or Celsius.
* Added the ability to control the duty cycle output of the ports. It currently runs the ports around 30 Hz and you can control the duty cycle from about 1% to 100%.
* Monitor the current supply voltage and if it goes below 11.6V turn off the dew heaters so as to not run down the battery. If the voltage recovers, turn the dew heaters back on automatically.
* Added scrolling to the display so I can display more data.
* Added the duty cycle, amount of time powered up and the current dew strap voltage to the display.
* Added a reset button.
* When you power up the dew controller display all the stored high and low values
* Added the function to clear the stored values if you power it up twice in a row with no external sensors hooked up.
* Added the ability to set the temperature offset above the dew point separately for each of the 4 dew straps and made the default 5°F.
* I added debug code to the serial output.
* I also redid mitaccio’s hand drawings to make them cleaner, more detailed and easier to understand.

There are quite a number of other small changes I made along the way too. I also cleaned up the source code and added documentation to it.

Parts list:

* Arduino Uno 3
* Arduino Shield (empty board for making easy connections to the Arduino)
* Ribbon cable
* Red LCD Display 2x16
* Ribbon cable/wire
* (2) 100K Ohm trim pots
* (4) 1K Ohm resistor 1/4W
* (4) TIP-120 Transistors
* (2) TO-220 heat sinks
* (4) Bullet diodes 5A
* DHT11 temperature/humidity sensor with resistor
* (4) Dallas One-Wire temperature sensor
* 330 Ohm resistor
* (4) 2.2K Ohm resistor 1/8W
* Arduino Project case
* (4) RCA F-Connectors
* (4) 1/8” or 3.5mm stereo jacks
* (2) 12ft 1/8” or 3.5mm stereo M/M cables (cut in half to give (4) 6ft cables)
* 5.5mm power connector
* 5.5mm & 5.2mm power plugs
* Fused cigarette lighter plug

The libraries and current source code can be found here:

Source Code: <http://michiana-astro.org/projects/DewCtrl/DewCrtl3_20120913.txt>

Required Libraries: <http://michiana-astro.org/projects/dewctrl/libraries.zip>

Video of the display operating: <http://s21.photobucket.com/albums/b251/Pauls72/Scope/?action=view&current=CIMG0002.mp4>

Original source code by mitaccio: <http://www.cloudynights.com/ubbthreads/attachments/5321008-Completed%20dew%20controller.pdf>

Original CN thread: <http://www.cloudynights.com/ubbthreads/showflat.php/Cat/0/Number/5321008/page/1/view/collapsed/sb/5/o/all/fpart/all/vc/>1

I have since mounted the DHT11 sensor inside the case. And cleaned up all the extraneous glue. What I display on the screen has changed slightly. The very first picture in the article is of the finished project.

I used the 12ft stereo cables cut in half to put the Dallas One-Wire temperature sensors on. So each sensor cable is 6ft long. I also color coded the sensors and their plugs by using colored electrical tape to identify each one. The Dallas sensors are not really one wire but are 3 wires, power, ground and data. You can run them in a parasitic mode and get the power from the data line and use only 2 wires. If you do this you cannot read the sensors as often. All the Dallas sensors are hooked up in parallel. Each Dallas sensor is manufactured with a unique address. The way the system works is the sensor with the lowest identity number controls the first output circuit and then the next lowest the second circuit and so on. The Dallas sensors ids are read when the dew controller is powered up or reset.

So in my case:

1. Blue
2. Yellow
3. White
4. Red

If you only use the white and red, 1=White and 2=Red.

I used heat shrink tubing and electrical tape on almost everything and the case became quite cramped. I ended cutting off one of the corner posts and using hot glue to hold it together.

The 4 - 2.2K resistors are used as a simple voltage divider circuit to feed one of the analog input pins to detect the dew strap input voltage.

The system tests all the sensors and adjusts the dew straps off/on every 18 seconds. It constantly checks the voltage. Voltage and time are constantly updated on the display.

My original cost was around $170 dollars plus shipping. I got parts from Sparkfun, Adafruit and Radio Shack, plus my parts junk box. I calculated if you ordered parts from Hong Kong or eBay and electronics from Jameco you could get the cost down to $100-$110 dollars.

If I did it again, I would use a DHT22 ($12) instead of the DHT11 ($6). It works across a larger range of temp/humidity and is more accurate. On the down side is it double the cost of the DHT11.

DHT11:

* Good for 20-80% humidity readings with 5% accuracy
* Good for 0-50°C temperature readings ±2°C accuracy

DHT22:

* Good for 0-100% humidity readings with 2-5% accuracy
* Good for -40 to 80°C temperature readings ±0.5°C accuracy

The other thing would do differently is to use a IIC/I2C/TWI/SPI Serial Interface module $4, to connect the display to the Arduino. This would reduce the number of data pins to drive the display to only two.

The one small trim pot is accessible through the front, so you can adjust the duty cycle using a small screwdriver without opening the case.

Right now I track High and low of each of the following:

* Dew Heater Input Voltage
* Each of the 4 Dallas temp sensors
* DHT11 temp
* DHT11 humidity
* Calculated dew point

I have played around with recording all the values to EEPROM every 15 minutes for up to 6 hours and the having up to 4 days recorded. But you have to connect it to your PC to read out the values.

I currently have ordered a RTC (Real Time Clock) module $6, so I can display the actual time, not just the time the dew controller has been running. I have also ordered an infa-red remote control and sensor $5. I plan to remove the to trim pots and control, the duty cycle, display contrast, the temperature offset of each dew strap above the dew point and the reading and clearing of the stored values using the remote.

If anyone has any future ideas for enhancements or wants to build one and needs some help, let me know.





