Astronomy Society of the Pacific -- Arduino Tinkering Project

Dew Alert!

Wolf Witt, October 2, 2016

Outline

General Project Objective

Build a astronomy-oriented gadget using Arduino-family micro-controllers, ideally based on Intel Curie (Arduino 101) or Intel Edison hardware.

Specific Design Objective

Observations of the night sky are strongly affected by ambient light and atmospheric conditions. As a result, observations tend to take place late evening or night, after thermally-induced atmospheric disturbances have settled down. At those times, however, temperature tends to drop and relative moisture content of the air increases. This moisture, if it becomes excessive, causes dew to form on equipment. When dew forms on optical surfaces -- the corrector plates of Cassegrain-type telescopes are especially susceptible -- observation sessions are forced to end, and telescope gear needs to be handled carefully, so that it dries properly and is not damaged by the moisture.

During an observing session, dew often sneaks up on astronomers. Therefore, astronomers would benefit from a small, portable device that continually monitors temperature and humidity, and indicates when conditions are about to degrade, so that astronomers can take corrective measures

before it's too late.

This design, called **Dew Alert**, implements such a device.

Feature Summary

- Continuous monitoring of temperature and humidity
- LCD status display
- Four dew alert states: safe, near, warn and dew
- State indication via easy-to-read green/yellow/red LEDs
- Ambient light sensor to automatically adjust brightness of indicator LEDs
- User configurable audible alarm
- User configurable alarm thresholds
- Built-in rechargeable battery with 8 to 10 hour run time and low-battery alarm

Basic Operation

1. Turn on the device by pressing the blue power button.

2. The display will show a welcome message similar to

```
** Dew Alert **
   V1.00
```

and the device will test its LEDs and buzzer (unless the buzzer is silenced via the setup menu; see

User Configuration below).

3. After the test, the display will show

```
READY! Press
Enter to Start.
```

To begin monitoring dew conditions, press the Enter button (either the red square button on the top right of the device or the round red button near the USB and power connectors). This press of the Enter button serves as a test of this button to make sure it is functioning. (If the button is not pressed, but five minutes have passed, the device will time out and advance to the next step anyway.)

4. After the Enter button is pressed, the display will show the current conditions, such as

```
T = 22.7C H= 61%
Td= 14.9C  SAFE
```

T is the current temperature, H is the current relative humidity and Td is the calculated dew point temperature. Assuming the total moisture content (absolute humidity) of the air remains constant, then the relative humidity (H) will reach 100% when the actual temperature (T) drops to the dew point temperature (Td). Absolute humidity may change independently, however, so the dew point is continually recalculated. (The temperature display defaults to degrees Celsius but may be changed to Fahrenheit; see *User Configuration* below).

When the air reaches 100% relative humidity, moisture will condense out of the air (i.e. fog will form), but dew will settle on surfaces before that point. Some surfaces are more susceptible to dew than others.

The "SAFE" indication on the display means that conditions are such that all surfaces should be safe, that is dew should not be forming anywhere. Other possible states are NEAR, WARN and DEW! Each state is distinctly conveyed by the LCD text display as well as the colored indicator LEDs and a buzzer. See *Indicator State Table* below for more details.

The Dew Alert device uses an ambient light sensor to dim the LEDs as darkness approaches. The LCD panel light may be turned off by pressing the Down button (the lower blue button) and turned back on with the Up button (upper blue button). If the dew state changes from NEAR to WARN or WARN to DEW!, the display light will automatically reactivate.

Furthermore, a transition from NEAR to WARN will be accompanied with an audible alarm as the built-in buzzer will intermittently beep, and WARN to DEW! will cause the buzzer to sound continuously. In either case, the audible alarm may be silenced by pressing the Enter button. (The buzzer may also be turned off completely through *User Configuration*; see below.)

Indicator (LED, Buzzer) State Table

	Green LED	Yellow LED	Red LED	Buzzer	Comment

SAFE	steady on	off	off	silent	Conditions are safe; dew should not be forming anywhere.
NEAR	off	slowly oscillating	off	silent	Dew conditions are approaching; be cautious.
WARN	off	on	slowly oscillating	intermittent beep (until canceled)	Dew may be forming on highly susceptible surfaces; optical surfaces may be affected soon.
DEW	off	off	blinking	continuous beep (until canceled)	Atmospheric moisture content is very high; dew is forming on most surfaces; optical surfaces will be impacted.

State Thresholds

A transition between adjacent states (SAFE/NEAR, NEAR/WARN, or WARN/DEW) may be triggered in one of two ways:

- The temperature margin (difference between current and dew point temperatures) crosses a threshold.
- The relative humidity crosses a threshold.

Towards increasing dew risk, crossing either threshold causes a state transition.

The default thresholds are as follows, although they may be changed (see *User Configuration* below).

	Temperature Margin (T-Td)	**Relative Humidity Level (H)**
SAFE -> NEAR	5.0 C	70%
NEAR -> WARN	3.0 C	80%
WARN -> DEW!	0.5 C	95%

While in NEAR, WARN or DEW! state, the display prefixes the state name with "T-" or "H-" to show whether the state whether the transition to that state was triggered by crossing the temperature or humidity threshold, respectively. For example "H-NEAR" means that the device considers the dew point to be near based on relative humidity. Similarly, "T-WARN" means that dew is imminent based on the difference between the current and dew point temperatures.

User Configuration

Temperature display units, state thresholds and buzzer behavior may be configured by the user, and this configuration will be retained while the device is off. To enter configuration mode, press the enter button while turning on the device and continue holding it for at least three seconds. After three seconds, the display will show

```
Dew Config Mode
[Hold for Reset]
```

If you continue holding the Enter button for another five seconds, the device's configuration will reset to factory defaults, and the display will show

```
Factory Reset...
Done.
```

However, if you release the button, you will see

```
* User Config *
Enter to Start.
```

Pressing Enter will change the display to

```
Show temperature
in Celsius.
```

At this point, the Up and Down buttons may be used to toggle the mode between Celsius and Fahrenheit. Pressing Enter again, saves the temperature mode and advances to

```
Temp Margin Thr.
WARN/DEW    0.5C
```

Again, Up and Down allow the value to be changed, and Enter advances to the next setting. The next few settings are:

- `Temp Margin Thr. NEAR/WARN`
- `Temp Margin Thr. SAFE/NEAR`
- `Humidity Thresh. WARN/DEW`
- `Humidity Thresh. NEAR/WARN`
- `Humidity Thresh. SAFE/NEAR`

The final setting controls the buzzer, and the display will show

```
Audible alarm
mode: sustained
```

where the mode may also be `limited` or `none`. The Up and Down buttons cycle through the modes.

- sustained: alarm will sound continuously unless canceled with Enter button
- limited: every few minutes, the alarm will sound for a few seconds unless canceled with Enter button
- none: the alarm never sounds

While in sustained or limited mode, a short beep is part of the device's power-on test sequence. In none mode, that beep is suppressed. (A low-battery condition will produce an audible alarm regardless of this setting.)

A final press of the Enter button will briefly show

```
Configuration
saved.
```

and then the device will transition to normal operation (see *Basic Operation* above).

Powering/Charging the Device

The Dew Alert device is powered by an integrated, rechargeable lithium polymer battery. Under normal conditions, a full charge should allow the device to run for about eight to ten hours.

The device may also be powered through the Arduino's USB port, but it may *not* be powered through the Arduino's 5.5/2.1mm power jack. That jack is instead used for charging the battery. The charge voltage must be 5V (*not* the 7 to 12V normally specified for the Arduino power jack), and the charger should have a 2A current capacity.

If the battery becomes discharged, the Dew Alert device will issue a low-battery warning on its display, and the alert buzzer will sound (regardless of whether audible alarms are enabled for the WARN and DEW states). At this point, the device must be powered off to avoid damage to the battery, then recharge the device before using it again. While charging, the device may appear to turn on partially. This behavior is normal and a limitation of the current implementation.

Bill of Materials

The following hardware was used for the prototype build:

- 1x Arduino Uno R3
  https://www.adafruit.com/products/50
- BME280 I2C or SPI temperature, humidity, pressure sensor
  https://www.adafruit.com/products/2652
- 1x Photocell (a.k.a. light-dependent resistor, LDR), 3k to 200k ohm (e.g. part 202403 at jameco.com)
- 10mm LEDs: 1x, green (e.g. part 2152104 at jameco.com), 1x yellow (2152121), 1x red (2152112)
- 1x Piezo Buzzer, 5V (e.g. part 2098523 at jameco.com)
- Negative (black background) 16x2 LCD display
  https://www.adafruit.com/products/399
- I2C/SPI character LCD backpack
  https://www.adafruit.com/products/292
- 1x 16mm illumiated pushbutton, blue, latching
  https://www.adafruit.com/products/1476
- 1x 16mm illuminated pushbutton, red, momentary
  https://www.adafruit.com/products/1439
- Square tactile button: 2x blue, 1x red
  https://www.adafruit.com/products/1010
- 1x LiPo battery, 3.7V, 1200mAh
  https://www.adafruit.com/products/258
- PowerBoost 1000 Charger
  https://www.adafruit.com/products/2465
- 1x General-purpose silicon diode
- 1x Smoke translucent enclosure for Arduino
  https://www.adafruit.com/products/821
- 1x Perfboad (e.g. https://www.adafruit.com/products/2670)
- Resistors, 1/4W, 5%
  - Series resistor for each LED: 220ohm
  - Pull-up resistor for each button: 10kohm
  - Series resistor for LDR: 10kohm
- 6x #4 1/4in pan-head, self-tapping screw plus washers
- Assorted copper wire, 22 gauge, solid

Arduino Pin Assignments

For Arduino UNO prototype build, pin and wire assignments are as follows:

(See also Dew_Alert.ino source code file.)

Pin	Function	Pin Mode	Wire
Digital 0			
Digital 1			
Digital 2	Enter/Action button	Input	orange
Digital 3	LCD back-light LED, red	Output, PWM	(actually via I2C)
Digital 4	Low-Battery Indication	Input(*)	white
Digital 5	Power button light	Output, PWM	white/blue
Digital 6	Enter/Action button light	Output, PWM	green
Digital 7	Up button	Input	violet
Digital 8	Down button	Input	light blue
Digital 9	Indicator LED Warn (yellow)	Output, PWM	white/yellow
Digital 10	Indicator LED Safe (green)	Output, PWM	white/green
Digital 11	Indicator LED Dew (red)	Output, PWM	white/red
Digital 12	Piezo buzzer	Output	white
Digital 13			
Analog A0	Ambient light sensor (LDR)	Input	gray
Analog A1			
Analog A2			
Analog A3			
Analog A4	I2C SDA (data)		blue
Analog A5	I2C SCL (clock)		yellow

I2C --> Temp/Humidity Sensor, LCD Display

(*) INPUT_PULLUP

Connection with PowerBoost Board

PowerBoost	Arduino	Wire	Comment
5V	5V	red	device power
GND	GND	black	device ground
EN	Power Button	green	device enable (on/off) (1)
LBO	Digital 4	white	low battery indication (2)
USB(Vbus)	PowerIn	yellow	charge power (3)

(1) power button either floats EN pin (device on) or shorts it to ground (device off)
(2) need diode between LBO and Digital 4, with cathode towards LBO; Digital 4 needs to be INPUT_PULLUP
(3) PowerIn is + of Arduino center-positive 5.5/2.1mm power jack; solder wire directly to + of this power jack

Charging must take place through the Arduino's power jack, but this jack may no longer be used to power the device directly. Voltage into this power jack must be 5V (*not* the standard Arduino spec of 7 to 12 V).

Initialization of Configuration ROM

A newly built Dew Alert device will not have valid configuration data in its EEPROM. As a result, it will not behave correctly until its EEPROM is initialized. To do so, simply follow the factory reset procedure described under *User Configuration* above.

State Change Test Mode

Testing the behavior of the device in each of the dew states -- SAFE, NEAR, WARN, DEW! -- is not easy to achieve since the sensor's temperature and humidity readings are not easy manipulate to force transitions across the entire range of states. To nevertheless test the indicator behavior in each state, the device provides a state-change test mode. This mode may be entered from the

```
READY! Press
Enter to Start.
```

display by holding the Enter button for five seconds. The device will then acknowledge test mode by displaying

```
State change
test mode...
```

and then the display will change to a standard operating display

```
T = 22.7C H= 61%
Td= 14.9C  SAFE
```

but the state (e.g. SAFE) is not derived from sensor readings but may instead be controlled by pressing the Up or Down button. For example, from SAFE, pressing Up will cause the state to change to NEAR, and while in NEAR, Down will prompt a return to SAFE. Pressing Up and Down together exits test mode to standard operating mode.

Development Environment, Firmware Code, Libraries Used

Original code was developed using Arduino IDE 1.6.9, on a DELL XPS 15 laptop running Windows 10.

The firmware code (*.ino) file is available here
https://github.com/robotobyte/DewAlert
and it relies on the following libraries:

Standard Arduino UNO libaries:

- Wire (for I2C)
- EEPROM (to save user configuration)
- math (for natural log function for dew point calculation)

Libraries for sensors and display:

- Adafruit_Sensor
  https://github.com/adafruit/Adafruit_Sensor
- Adafruit_BME280
  https://github.com/adafruit/Adafruit_BME280_Library
- Adafruit_LiquidCrystal
  https://github.com/adafruit/Adafruit_LiquidCrystal

Wolf's libraries:

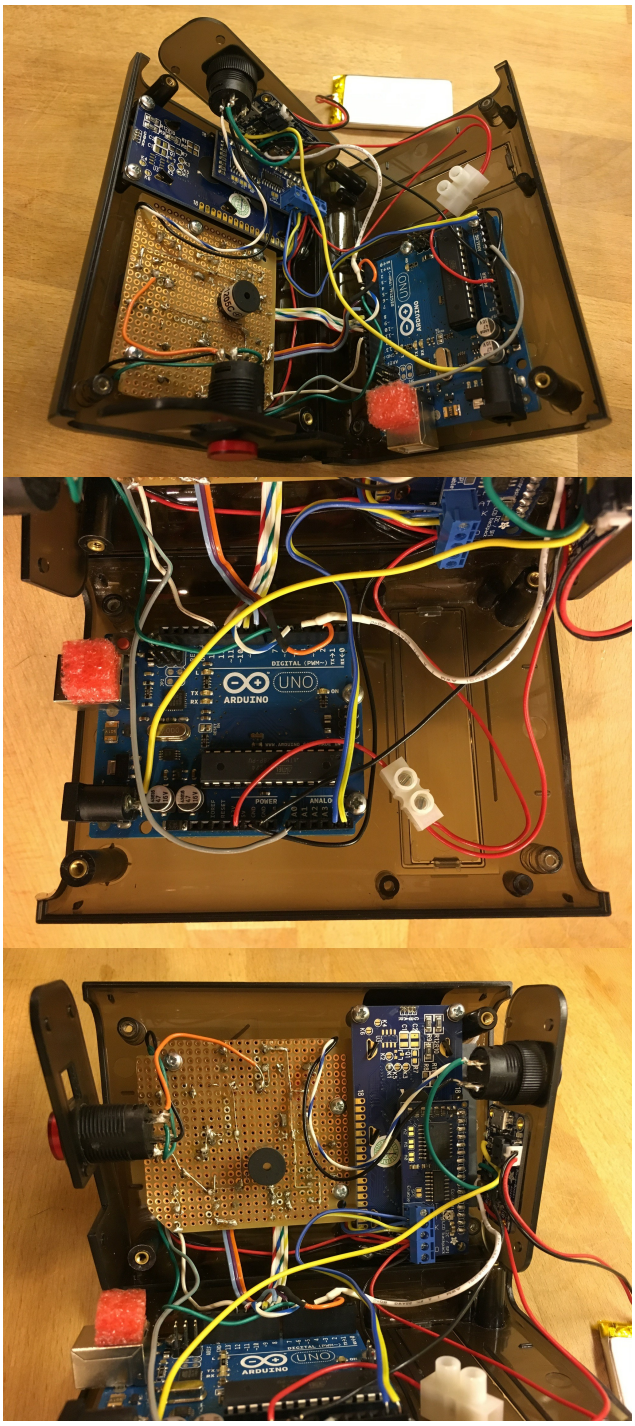
- CwwButton: easy interface to a button with debounce function
  https://github.com/robotobyte/CwwButton
- CwwElapseTimer: general purpose timer
  https://github.com/robotobyte/CwwElapseTimer
- CwwLedController: many functions for controlling an LED (or buzzer)
  https://github.com/robotobyte/CwwLedController
- CwwFilterHysteresis: filter to smooth out sensor values jittering around a threshold
  https://github.com/robotobyte/CwwFilterHysteresis

At the time of this writing, the version of the Dew Alert code is **V1.00 Beta 06**.

Future Devlopment (in no particular order)

- Add diode/fuse protection to charging circuit (i.e. protect against wrong polarity, excessive voltage and current).
- Add circuit that automatically powers off the device if a low-battery condition is sustained for some amount of time.
- Improve charging circuit, so device does not partially power up (even while off) when charging power is applied.
- Tweak the hardware, so that the red backlight of the LCD panel is not controlled via I2C (which just provides on/off function) but can be connected to a PWM-capable digital output pin (e.g. Arduino digital pin 3), so that its brightness may be controlled based on the reading from the ambient light sensor.
- Instead of the BME280 temperature and humidity sensor, use the more precise Sensiron SHT31-D sensor (https://www.adafruit.com/products/2857).
- Implement more accurate (and more complex) dew-point formula.
- Port the design from an Arduino Uno to an Intel Curie-based Arduino 101. Use the Arduino 101's built-in Bluetooth capability to create a link to a smart phone, so that environmental data may be displayed and alarms may be communicated via the smart phone. The existing software should port cleanly to the Arduino 101 with the exception of tweaks that may be needed for the routines that store user configuration data in, or read it from, EEPROM.
- Design (and 3D print?) a custom case that provides a better fit for the device's internals and allows for a larger battery.
- Replace the ugly, ugly perfboard-based circuit board with a nicer, custom printed circuit board.

Pictures of Internals (Messy Prototype Build)

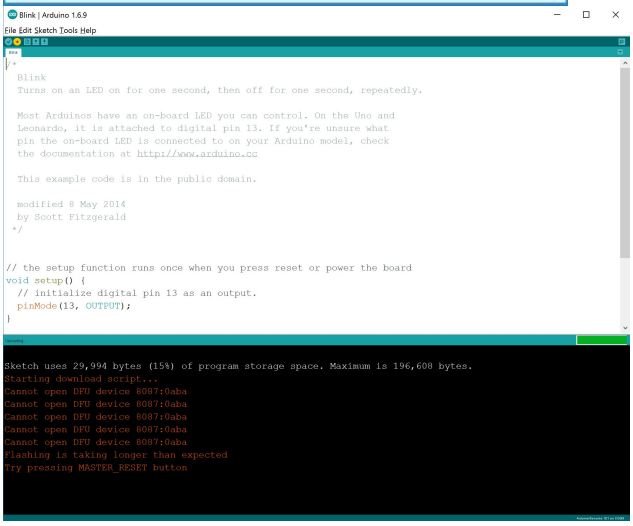


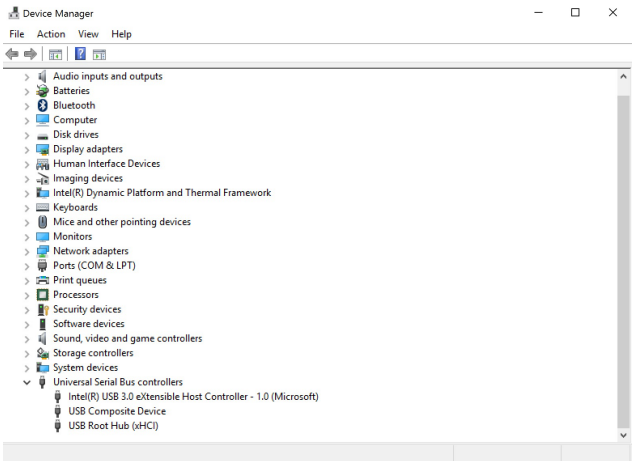
Arduino 101 Difficulties

The original plan was to build a device out of an Arduino 101, Intel's Curie-based micro-controller

board. However, the 101 board refused to work with any of three Intel/Windows-based laptops I tried. In all cases, the Arduino IDE reported the Arduino 101 board properly, but attempts to upload a sketch failed with "Cannot open DFU device 8087:0aba". Attempts to press the Master Reset button at strategic times did not improve the situation. Some Web resources suggest that a USB 3 interface is required, but all laptops I tried did indeed have USB 3 capability. See relevant sample screen shots below.

The Intel Curie-based Arduinio 101 is an attractive board, but this nonsense with DFU errors spoils the experience and stands in the way of product development. Similar to an Arduinio UNO, these devices just have to work or further market acceptance and penetration. In my case, I had a choice to continue to debug (waste time with?) the Arduino 101 or build something with an Arduino UNO. After a sufficient amount of frustration, I chose the latter.




Device Manager
File
Action
View
Help
Audio inputs and outputs
Batteries
Bluetooth
Computer
Disk drives
Display adapters
Human Interface Devices
Imaging devices
Intel(R) Dynamic Platform and Thermal Framework
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Ports (COM & LPT)
Print queues
Processors
Security devices
Software devices
Sound, video and game controllers
Storage controllers
System devices
Universal Serial Bus controllers
Intel(R) USB 3.0 eXtensible Host Controller - 1.0 (Microsoft)
USB Composite Device
USB Root Hub (xHCI)