Nano Environment - an Arduino based weather station 2023

This design predates the latest version of the AAG CloudWatcher, which now includes SQM, pressure and humidity measurements. I also completed this design before the similar Pegasus Uranus Meteo was announced. It uses an original Texas semiconductor sensor which I used 15 years ago for my commercial darkroom meter designs. It is very sensitive and I have since discovered it is now the mainstay of several SQM meters. The only advantage over the otherwise excellent AAG unit (of which I have several) is the fun of making your own and the ability for handheld measurements.

This is not an easy design, on account of the pedantic nature of some of the sensors. The rain sensor is sensitive, but tricky to solder to the wiring. It is quite delicate and I am not certain of its ongoing robustness to extended exposure in the environment. The heater (and its thermistor sensor) permit controlled evaporation of condensation of the sensor surface. The MLX90614 has different readout modes, which are not trivial to set, and is sensitive to electrical interference.

The project comprises hardware, firmware, and an ASCOM driver. The hardware is based around an Arduino Nano, though other units can be used. It is housed in a plastic enclosure, and in my implementation, has a 7-segment LED display and a pushbutton, for manual measurements. It can be powered from its USB cable, though this is insufficient to power the rain sensor heater.

The sensors provide several functions:

MLX90614. This is the IR temperature sensor found in forehead thermometers. They were scarce during COVID, but are more easily found now. This faces upwards to the sky and the difference between the IR sky temperature and the ambient temperature is the basis for estimating cloud cover. My cloud cover calculation is self-calibrating, and is arguably more sensitive than some other algorithms.

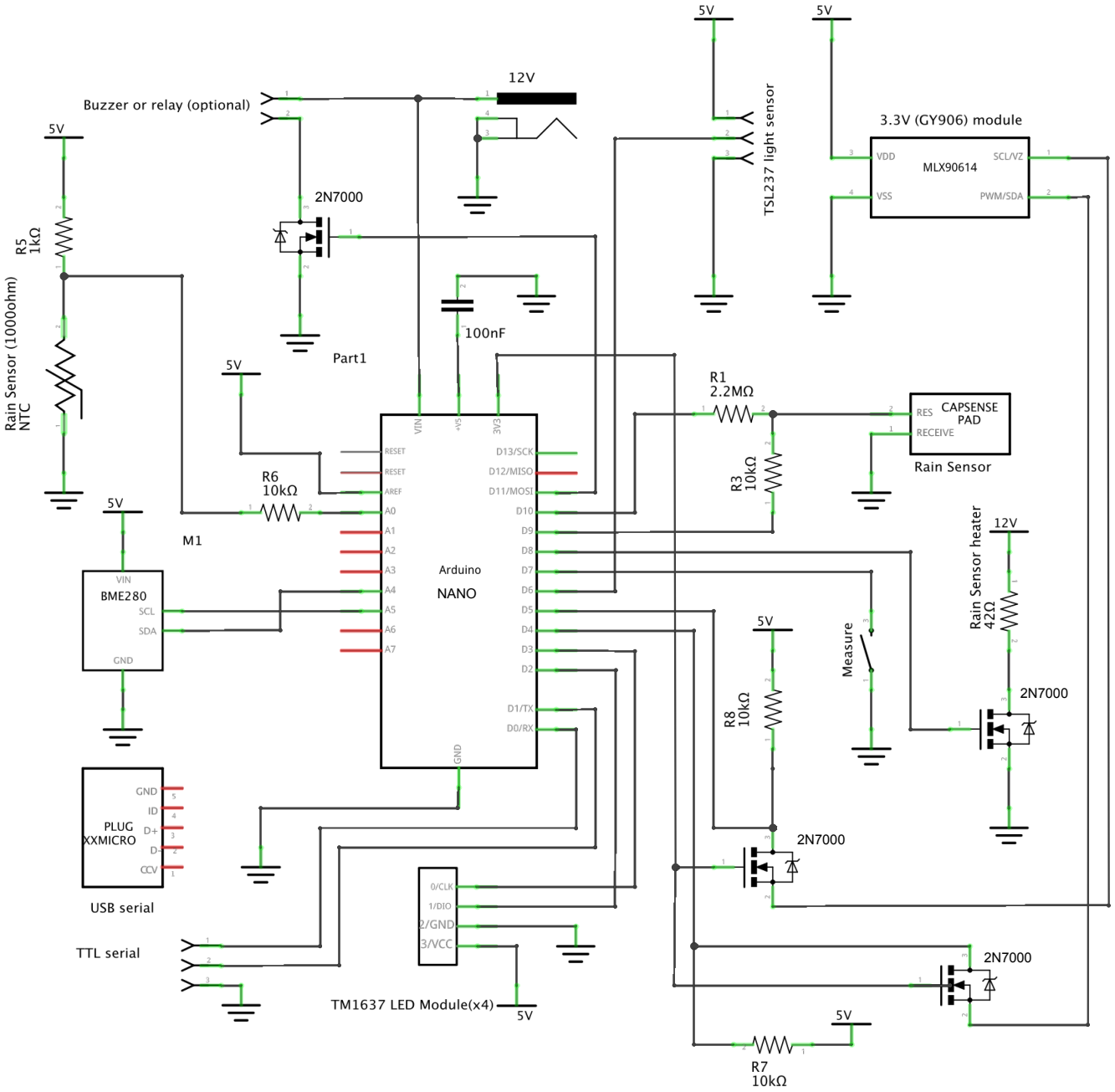
TSL237: This is a three-pin light to frequency converter chip. The frequency is sub-hertz at nighttime light levels, and is easily measured with an Arduino. It requires optical filtering with a small UV/IR filter for more accurate SkyQualityMeter readings and needs shielding from any extraneous light (or board LEDs). There are alternative similar sensors, of varying sensitivity. We need the most sensitive. This is used to calculate SQM.

Thick Film Rain Sensor. RC-SPC1KA: This has a capacitive element for measuring moisture and a built in heater with a thermistor sensor. The capacitance is measured directly by the Arduino, which manages selective heating at power up, and when moisture is detected. It is self-calibrating to account for humidity extremes. It is worth looking for alternatives as this unit is quite expensive and tricky to mount.

Bosch BME280: This tiny sensor is at the heart of many environmental projects. It measures temperature, humidity, and pressure. It needs to be physically positioned so that its temperature measurements are not affected by the other electronics. The dew point is a derived calculation.

TM1637 LED module: Simple to use red-LED display. When the measurement button is pressed, it cycles through all its measurements. Alternatives abound, but will require their own libraries and code for writing strings and numbers. It is optional and the relevant Arduino code can be deleted, as required.

2N7000: These power transistors are used in various places as switches for heaters/buzzers.



Circuit Notes.

The NANO is a 5V device and the MLX temp sensor uses I2C communication at 3V3. Two 2N7000 transistors convert the logic voltage levels between the module and the Arduino. The MLX is typically bought on a small module board, which does its own 3V3 conversion from the 5V power supply. Keep the SCL/SDA wires short and filter the power lines with a capacitor.

The BME sensor is also bought as a tiny module and has electronics to work with 5V power and connections. It is connected directly to the NANO with SCL / SDA (clock and data) serial pins. It should be positioned where it is not directly heated by the electronics.
The rain sensor is capacitive. It is measured by charging and discharging the capacitor, through a 2.2M resistor and noting the time for the decay on another NANO pin. The sensor heater is a 42 ohm film resistor and is turned on and off with another transistor. Its integrated thermistor (of

about 1K) is sampled by an analog pin and resistor divider. The heater is only on when needed and is turned off during measurements or if it exceeds ~25 °C.

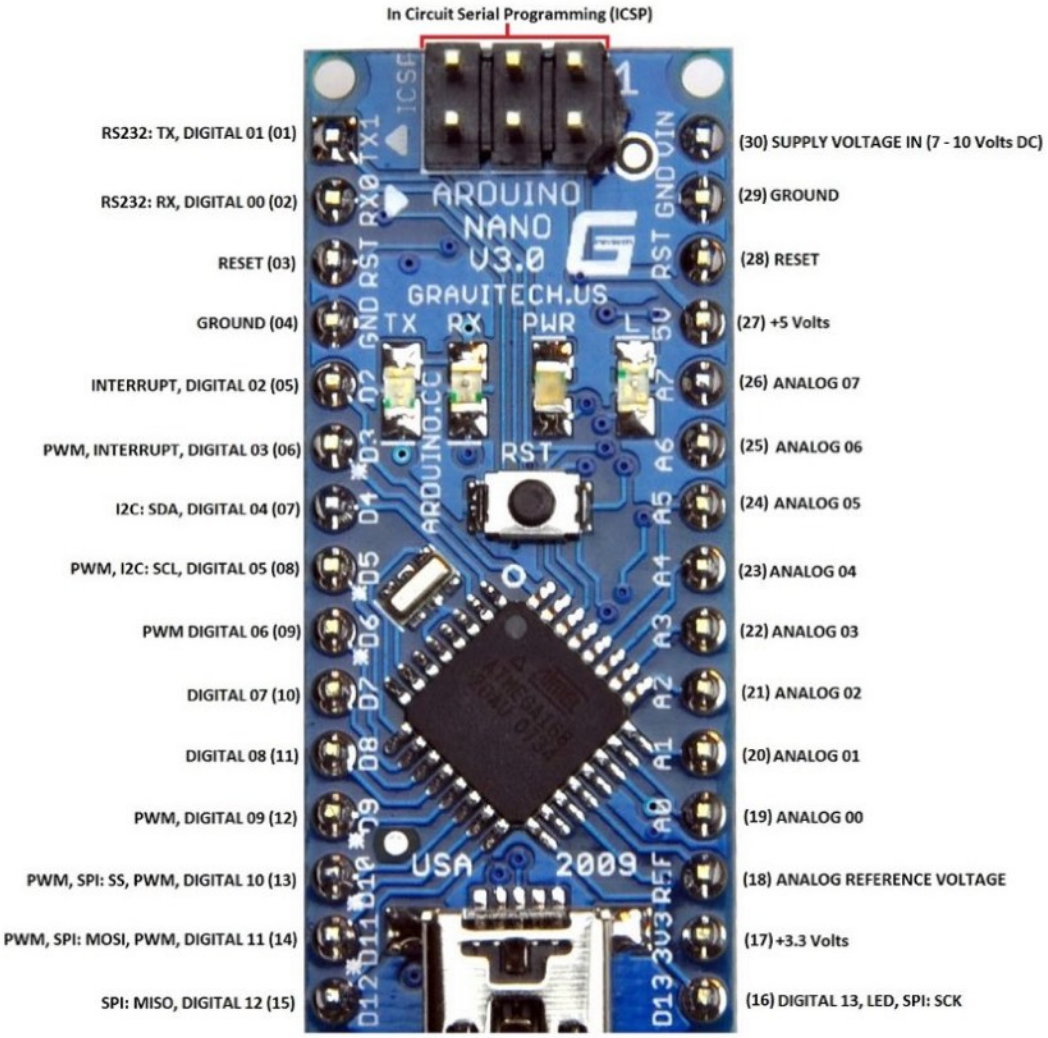
The TSL light sensor is simple to connect up. It has 5V/0V power connections and a single output pin, which broadcasts a square wave of varying frequency. Note the sensors orientation to face towards the light. Be aware that some black plastics are transparent to IR.

The optional LED module is powered by 5V and communicated with a pair of clock/serial connections.

An optional 12V buzzer is used to indicate unsafe conditions and is powered via another power transistor. The 12V is applied to VIN pin, and is regulated on the NANO board as an alternative to being powered from its standard USB connection.

Serial ports
A second serial port is used for diagnostic purposes. I only use one in operation.

There are several NANO versions and clones - with similar pinouts. They are labelled on the board and it is worth checking them carefully before soldering.



Arduino Nano V3 - Pin Description

www.CircuitsToday.com

Software:
I initially used Arduino's own free IDE to develop the code for the board, but later moved over to Visual Studio, which was required to develop the ASCOM driver. To program the board and link to the libraries, you will need the Visual Micro extension for VS. It has a 30-day grace period before a small charge is required for ongoing use.

The Arduino software uses C++ and makes extensive use of libraries for the MLX and BME sensors. Some MLX libraries use a different version of the module, which is permanently configured for serial communication. The software takes the average of multiple readings to remove noise.

The general principle is that the Arduino reads its sensors, manages the rain sensor heater and transmits a string of data values every few seconds over its serial port. It is not necessary to read all the sensors each time, the most time sensitive ones (like rain) are sampled more frequently. It compensates the sky temperature for the ambient conditions and calculates cloud cover from the result. It transmits temperature (from BME), humidity, pressure, adjusted sky temp, ambient temp (from MLX), sky magnitude (SQM) and cloud cover.

The ASCOM driver is written in C# and reads the serial string and responds to driver calls. It also manages some trim values for the sensors to match other module readouts. The ASCOM driver also calculates the dew point from the temperature and humidity.

Construction and Use:
I found an IP65 rated box and then cut holes in it. I sealed all the upper holes with glue and varnish and then placed the BME sensor in the base, facing downwards, with ventilation holes. There is a lot of circuitry and I soldered the Arduino to a small prototype board for all the connections and small components. It is quite fiddly to solder, on account of the density of connections and wires. I used thin solid core wiring for hookups.

Once assembled and programmed, simply connect up power and use the button to run through the environmental data. Some data is refreshed quickly (like temperature) but others, like SQM and cloud cover are updated less frequently. In full darkness, the SQM value should be about 20 mag. Putting a damp finger on the rain sensor should immediately set the rain alarm on (buzzer and display). It should trigger the sensor heater and after removing your finger, it should reset in a few seconds. I borrowed an SQM and calibrated mine against it.

As I already have the latest AAG CloudWatcher permanently mounted in my observatory, I use this unit for field use.