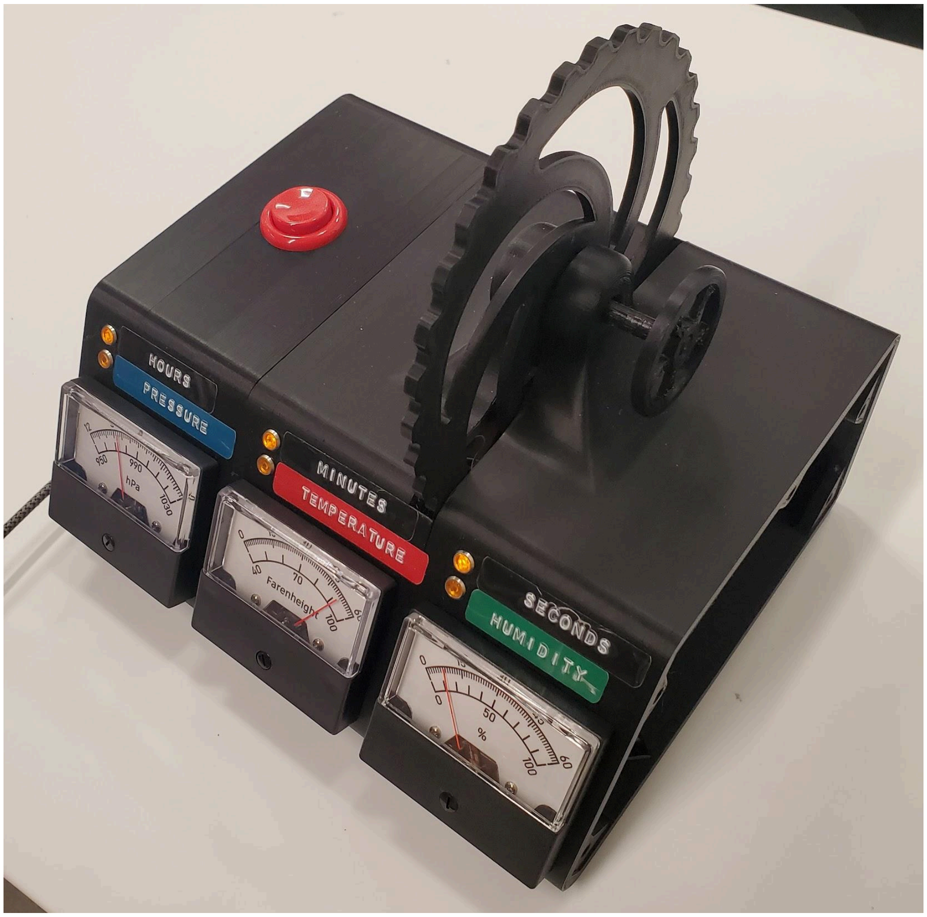
# Analog Voltmeter Time, Pressure, Temperature, & Humidity Display Clock

San Francisco State University



Leonard Chau

December 2024

lchau@sfsu.edu
www.leonardchau.com

# Project Overview

The device I chose to build is a clock combined with a pressure, temperature, and humidity monitor using analog voltmeters for an engaging and unique display. The analog voltmeters feature a custom legend that indicates the current hour, minute, and second, or pressure, temperature, and humidity on a graduated scale. I was inspired by the idea of using analog displays to show time, which has a classic and visually engaging aesthetic. To further challenge myself, I integrated environmental monitoring features, making the device multi-functional. All the components and encoder knob assembly are securely fastened to a custom 3D-printed enclosure made of PLA on an Ender 5 printer. Time adjustments are made using a homemade rotary encoder and a button. The display toggles with a short press of the button.

# Physical Design

For the physical design of my device, I aimed to create something that is large and unique—a device that would instill curiosity at first glance. The analog voltmeters and vintage embossed labels aim to provide a sense of nostalgia. The oversized rotary encoder is a deliberate design choice, serving both functional and educational purposes. The method of function of an incremental rotary encoder can now be felt, seen, and heard. It offers a satisfying, hands-on experience and acts as a demonstration tool to illustrate the inner workings of a rotary encoder.

The analog voltmeter's original 0-30V graduated display was modified with custom-designed and printed legends that indicate hours ranging from 0-12, minutes and seconds ranging from 0-60, pressure ranging from 950-1030 hPa, and relative humidity ranging from 0-100%. These values were chosen as the range at which this device is estimated to function. The voltmeters, originally rated for 30V, had to be further modified to operate at 5V by removing their internal resistors and installing adjustable potentiometers for calibration.

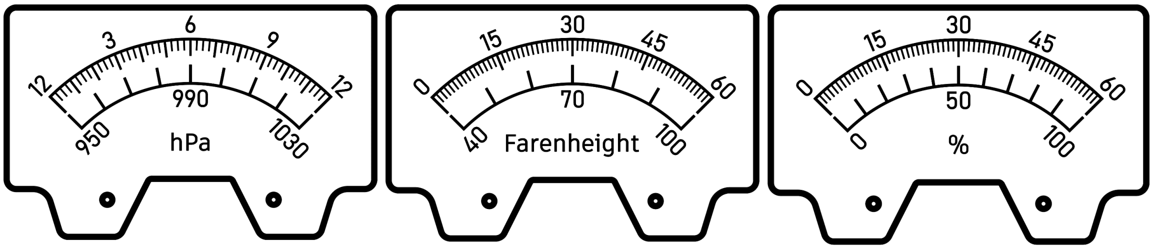


Figure 1: Pressure/hour, temperature/minute, humidity/second legends

# Function Flowchart

A DS3231 real-time clock module (RTC) is used to keep accurate time and maintain the time even when the Arduino is fully powered off, thanks to an external watch battery. It communicates with the Arduino via I2C protocol, which makes integration with the Arduino seamless. The current time provided by the RTC is obtained constantly and stored as variables in the Arduino. The BMP180 pressure sensor also communicates via I2C, and the current pressure is stored as a variable. The DHT22 humidity sensor operates via pulse-width modulated input, which is handled by a library, and its variable is also stored.

The time and environmental variables are converted to integers in the range of 0 to 255 to provide pulse-width modulated outputs for the voltmeters. When the display mode is set to TIME, the voltmeters display the current hour, minute, and second at quarter-second intervals to provide for a sweeping second hand, and the HOURS, MINUTES, and SECONDS LEDs remain on. When the display mode is set to PTH (pressure, temperature, and humidity), the voltmeters display the current pressure, temperature, and humidity, and the PRESSURE, TEMPERATURE, and HUMIDITY LEDs remain on.

The display mode is toggleable via a short press of a user-interaction arcade button. Upon a long press of the arcade button, a TIME CHANGE mode is entered, in which the rotary encoder is used to select the new HOUR, MINUTE, and SECOND. This is indicated by a blinking corresponding LED and isolated operation of the selected voltmeter. A new value is confirmed by pressing the arcade button until the last parameter, SECONDS, has been updated, returning to TIME display mode.

The rotary encoder operates on two input signals that are treated as interrupts, performing interrupt service routines (ISRs) when their signal values change, providing for a responsive and effective incremental rotary encoder.

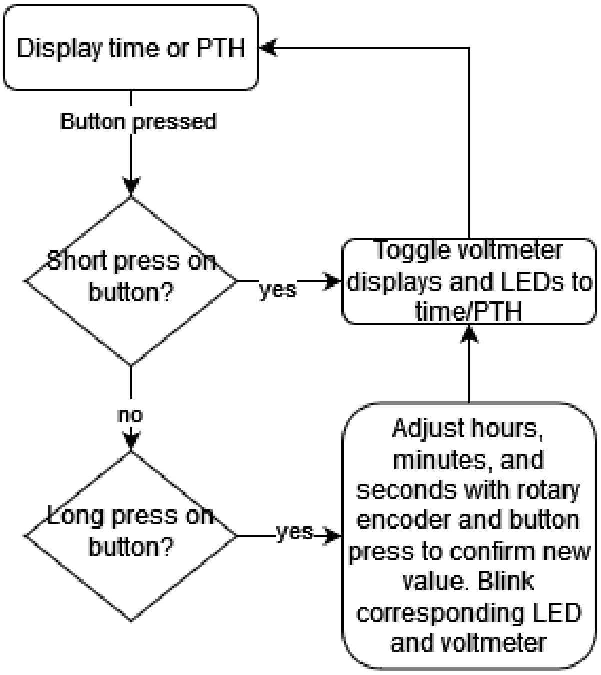

Display time or PTH
Button pressed
Short press on button?
yes
Toggle voltmeter displays and LEDs to time/PTH
no
Long press on button?
yes
Adjust hours, minutes, and seconds with rotary encoder and button press to confirm new value. Blink corresponding LED and voltmeter

Figure 2: Flowchart of function

# Components and Materials

The device is built using the following components:

- One Arduino Nano (microcontroller)
- Two limit switches (used for rotary encoder input)
- One arcade button (for toggling the display and time adjustment)
- One DHT22 sensor (for temperature and humidity sensing)
- One BMP180 sensor (for pressure sensing)
- Six 6V LEDs (for illumination and visual feedback)
- One DS3231 RTC (Real-Time Clock) module (for accurate timekeeping)
- Three analog voltmeters (to display time and environmental data)
- Three 5kΩ potentiometers (for analog voltmeter calibration)
- One breadboard (for soldering all the components to)
- PLA plastic (for the 3D printed housing)
- Five M6 screws (for assembling the 3D printed housing)

# Schematic

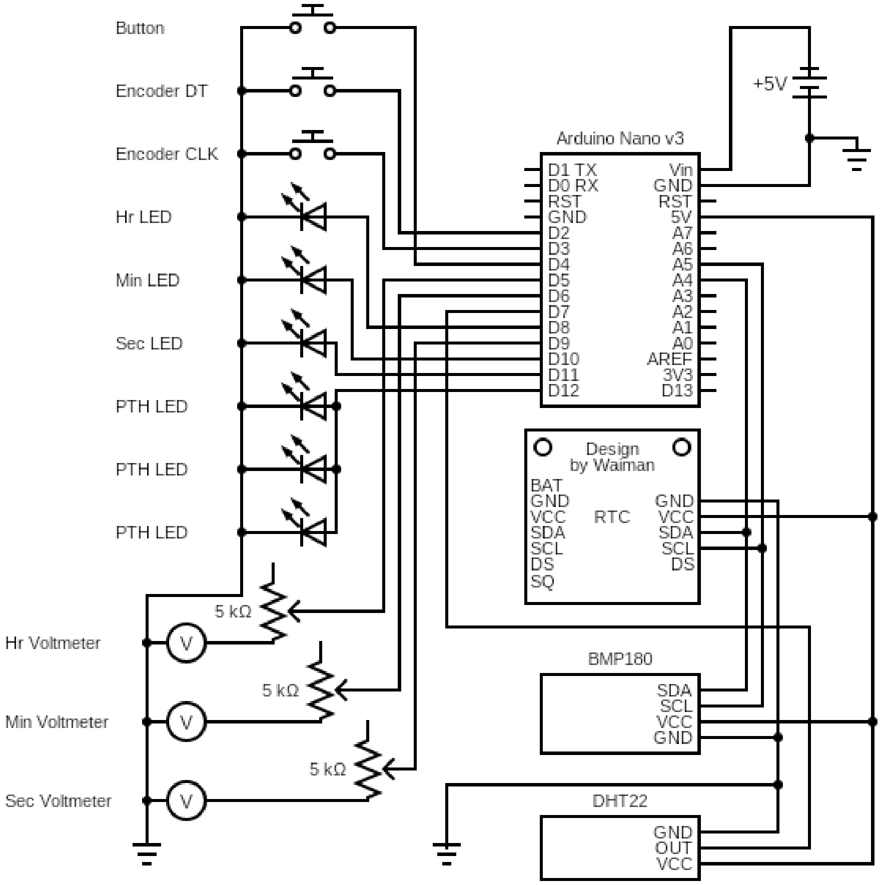


Figure 3: Wiring schematic of device

# Development and Testing Process

I developed and tested the project in stages to ensure functionality at every step. Initially, I assembled and tested individual components on a breadboard. This modular approach allowed me to troubleshoot and confirm that each component worked as expected. Once I was able to read, process, and manipulate the data from all the sensors and inputs, I merged all the separate programs into one cohesive script. This combined code allowed the device to perform all its intended functions seamlessly.

# Project Outcome

The final product closely aligns with my original vision. However, reflecting on the process, there are several improvements I would consider if I were to redo or continue this project:

1. Data Storage and Connectivity: While the display of real-time environmental data is visually appealing, it would be more useful if the data could be stored for later analysis. Adding wireless connectivity (like Wi-Fi or Bluetooth) or local storage (like an SD card) would allow for long-term data tracking.
2. Enhanced Rotary Encoder Functionality: Currently, the rotary encoder's only function is to adjust the time. It would be beneficial to introduce additional features for when the encoder is moved during its default display state. Possible features include controlling lighting intensity, adjusting volume, interacting with kinetic art, or activating a countdown timer.

# Lessons Learned

Completing this project taught me how straightforward it can be to integrate multiple components into a single program to achieve a larger, more complex function. Completion of a successful project with code functioning as intended provides a rewarding and confidence-boosting experience. I also learned the value of using powerful tools to solve coding challenges. For example, leveraging AI tools like ChatGPT provided valuable guidance in debugging and optimizing my code. This project has given me the confidence to take on more ambitious projects that incorporate diverse inputs and outputs into a unified system. I am all the more prepared and excited to design and create new projects involving Arduino in the future.