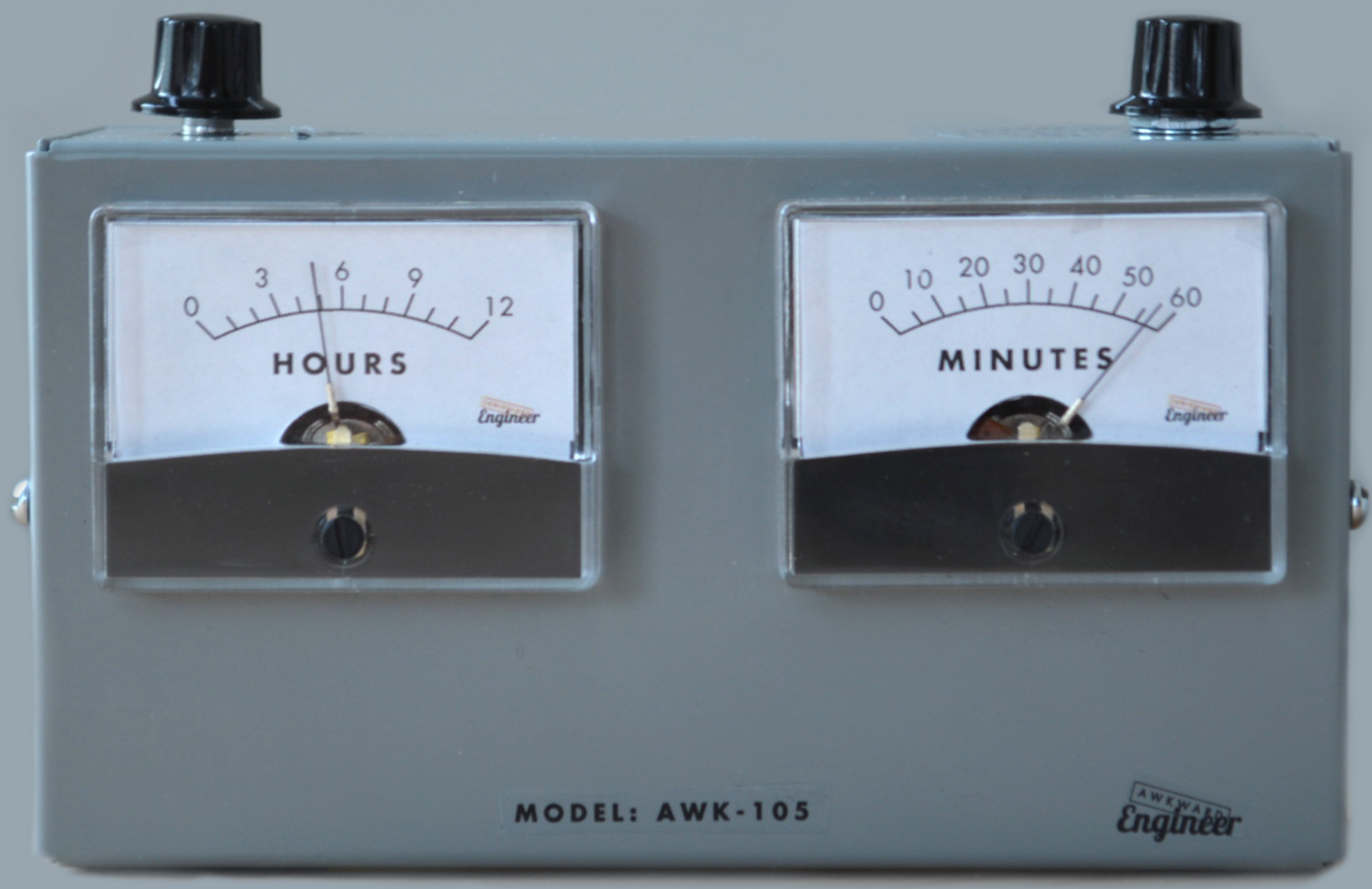
AWK-105

"Analog Voltmeter Clock"

www.awkwardengineer.com

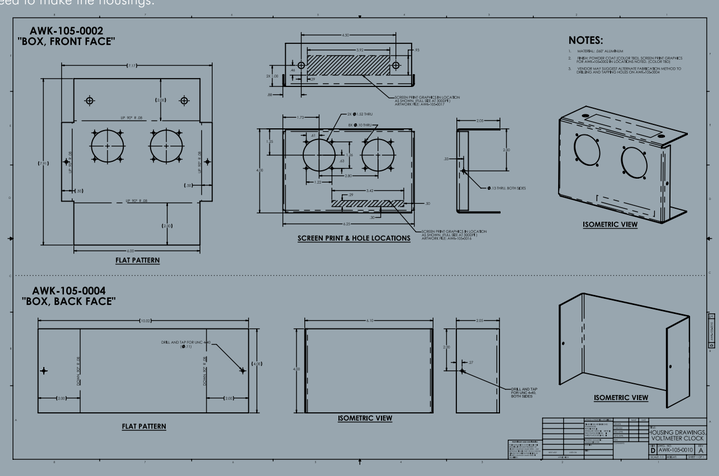


Dual analog meters tell time, currently showing 4:57.

(Almost 5 o'clock somewhere.)

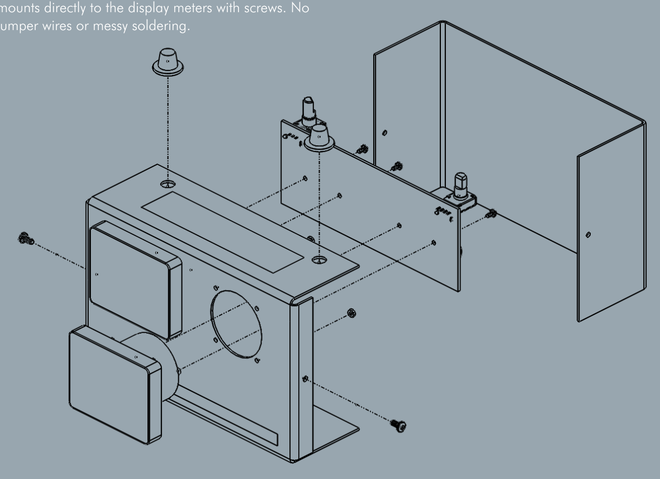
Design for Manufacture

Detailed drawings contain all the information a manufacturer will need to make the housings.



Design for Assembly

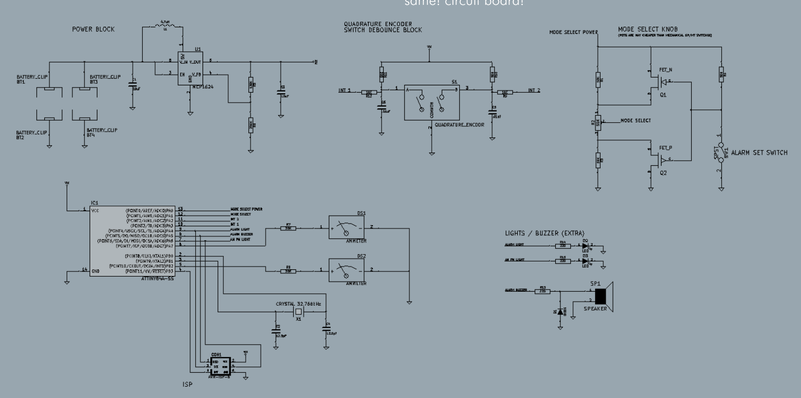
The circuit board, designed with right angle knobs, mounts directly to the display meters with screws. No jumper wires or messy soldering.



Schematic Capture

Built around an Atmel ATtiny84 microprocessor, and an MCP1634 boost converter, the Clock draws an average of 300uA off two AA batteries.

Both the MODE SELECT knob position and an ALARM SET switch are read from a single analog input. The switch feature is optional. Leaving some components unpopulated creates an entirely new design with the same! circuit board!



Firmware

Prototyping was done in an Arduino environment, but to take advantage of all the low power features of the ATtiny84, low level control of the registers was needed.

The code is interrupt driven, meaning that microprocessor spends most of its time asleep. One set of timer registers controls the PWM outputs to the meters and the other counts milliseconds. The "SET TIME" knob is an encoder that generates interrupts, but the "MODE SELECT" potentiometer has to be polled.

```
int main (void)
{
    mode_pointer = &mode_time; //assigns the mode_pointer initially to time mode

    pwm_init();
    timer_init();
    analog_init();
    pinchange_init();
    power_register_init();
    sei(); // global set enable interrupts

    while(1)
    {
        PRR &= ~(1<<PRADC);     //turns on the ADC comparator
        PORTA |= (1<<PA6);      //turn PA6 (pin 7) on to power the potentiometer
        ADCSRA |= (1<<ADSC);    // starts AtoD conversion by flipping ADC Start
                                //  Conversion bit in AD Control and Status Register A

        while(ADCSRA & (1<<ADSC));  // loops while waiting for ADC to finish

        PORTA &= ~(1<<PA6);     //turn PA6 back off to conserve power
        PRR |= (1<<PRADC);      //shuts down the ADC and comparator


        (*mode_pointer)();      //uses a pointer to call the function for the specific mode

        sleep_cpu();
        //just hang out and wait for interrupts
    }

    return 1;
}
```

Circuit Board Layout

Basic board layout was done using KiCad. Footprints for the knobs and battery clips had to be made from scratch.

Circuithub.com has excellent tools for sourcing components and alternates. We're using them to make and assemble prototypes!