Thank you for ordering an ESPixelClock and helping me bring this project to life! Here are a few things to get you started…

If you ordered a board with an ESP32 pre-soldered, it has already been flashed with the code located here: <https://github.com/mattncsu/ESPixelClock/tree/master/examples/ESPixelClock>

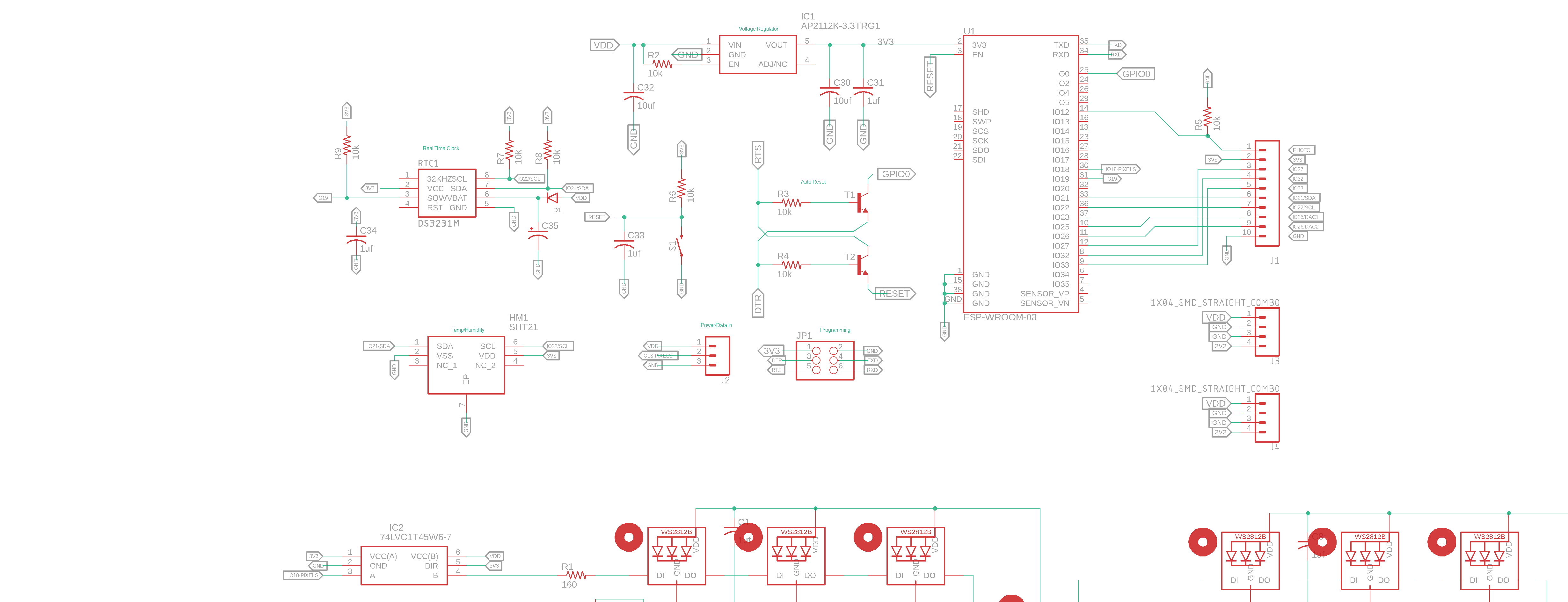
On power up, the clock will attempt to connect to a WiFi access point which will fail after 8 seconds since none is configured. The clock will then spin up an access point with the SSID ESPixelClock xx (where xx is the last two digits of the MAC address) with the password “espixelclock”. Connect to the hotspot and then open up a browser to <http://192.168.1.1> or <http://192.168.4.1> should the first address not work. Use the “Wifi Config” tab to set the SSID. After hitting submit, the clock will reboot and try to connect to your access point. Use your router’s network map page, an app like Fing to scan your network to find the new IP address or connect a usb-serial adapter to the TXD and GND pins to listen to debug messages from the clock.

For programming, you can compile a sketch and upload it using the Firmware upload page (http://<ip address>/update) or upload it over serial. Any USB-serial adapter with RTS/DTR pins will work. I personally use one like this <https://www.amazon.com/CP2102-Multi-Function-Serial-Converter-Compatible/dp/B08246HWJ4/>.

For the bare board, simply connect the microcontroller of your choice to the D\_IN pad. Board can be powered from any of the 5V pins and may also run on 3.3V.

Specifications:

* LEDs: 86 x WS2812B-mini 3535 addressable LEDS
* RTC: DS3231M at I2C address 0x68 (silkscreen has typo) with capacitor backup
* Environment Sensor: SHT21 at I2C address 0x40



See <https://github.com/mattncsu/ESPixelClock/tree/master/hardware> for full schematic/layout