GPS clock

Build an GPS-based clock using surface mount components.

Bruce E. Hall, W8BH

Introduction

I recently breadboarded a GPS clock and it works great. With only 4 components (power module, Blue Pill, display module, and GPS module) you can build a very useful time-keeping device. It is a straightforward exercise to solder them together and put them in the enclosure of your choice.

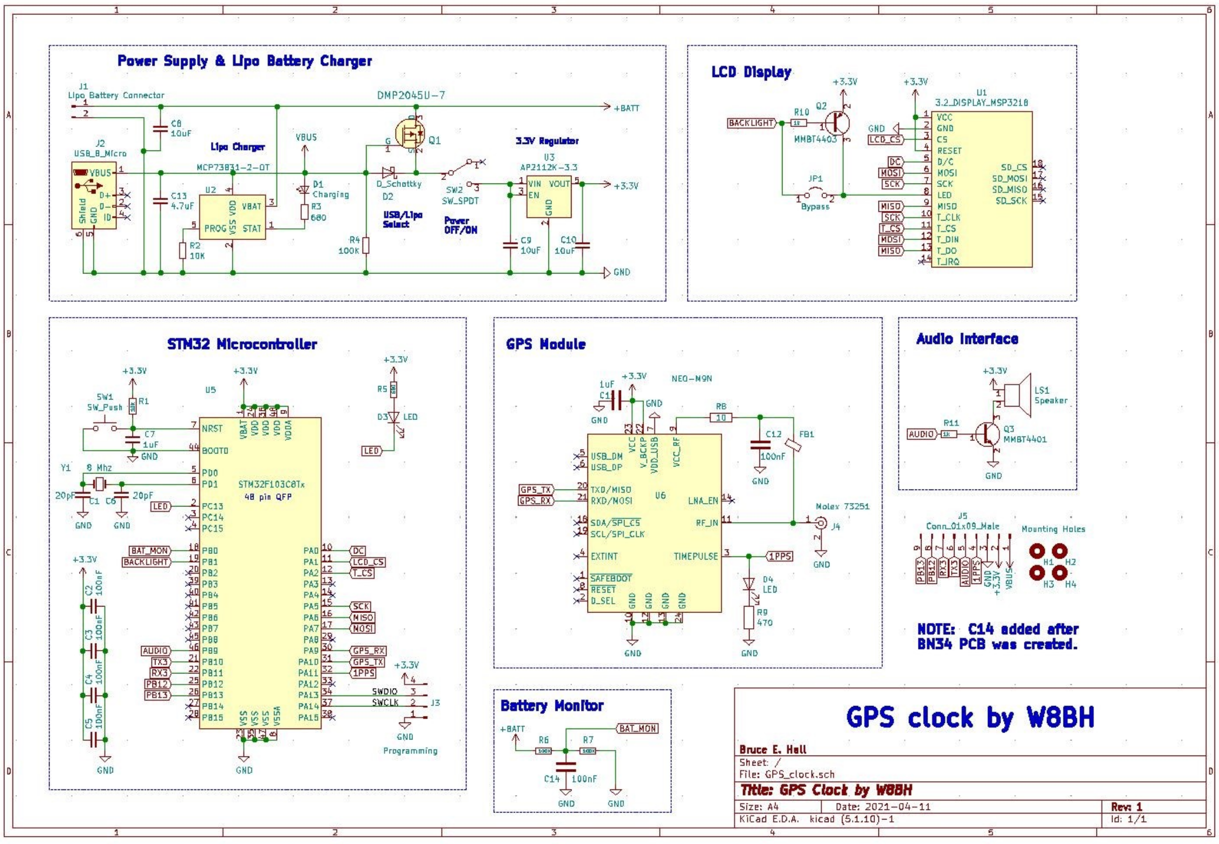
For my project, I eschewed pre-built modules and built the clock from scratch, using SMD components. This is a bit crazy and not the easiest choice: the cost of the project is tripled, and so is the complexity of the build. So, if you are sane and reasonably-frugal person, you can stop reading here and continue your normal and happy life.

So why did I build it from "scratch"?

1. I wanted to be able to build a battery-operated, hand-held device with TFT display and plenty of board space for project-specific parts. This clock fits the bill.
2. The pre-built modules work great, but don't do exactly what I had in mind. For instance, I could not find a good 3.3V power supply module that provided automatic switching between battery and 5V input while keeping the battery charged.
3. I wanted a personal challenge. I have made created a few of my own projects, made a few PCBs, and have done simple SMD work. This project would test all of those skills.

If you are still interested, then you must be an advanced builder: someone who knows microcontroller hardware, who can build from a schematic, and who doesn't want or need a lot of explanation. So, I will keep my notes brief, highlighting some of the interesting "gotchas" I encountered.

The Schematic



The schematic consists of 6 sections: power supply, microcontroller, display, GPS, Audio, and battery monitor. The link to the full-size schematic is here: GPS CLOCK SCHEMATIC. The associated bill of materials is here listed at the end of this article. Each section is discussed below

Power Supply

The microcontroller and GPS modules both require 3.3V. This voltage allows us to use a 3.7V single-cell LIPO battery as a power source, but just barely: with only 0.4V of headroom between the nominal LiPo output and the 3.3V regulated output, we have to be very careful with component selection. Many voltage regulators require 1 volt difference (or more) between their inputs and output. A "low voltage dropout" LDO regulator is required. I am using the AP2112.

A standard way to switch between power sources is to isolate them with diodes. The cost is a small voltage drop across the diode. For a standard silicone diode, this voltage drop is 0.6-0.7 volts, too much drop for our battery. Schottky diodes have a lower voltage drop of 0.2 volts. This is much better, and OK for 5V input, but marginal for battery input.

Instead of a diode, this circuit uses a P-channel MOSFET (Q1), which is turned on when the battery is present, presenting very low resistance and essentially no voltage drop. When 5V volts is applied, however, the gate of Q1 is high, Q1 is turned off, and the internal body diode is reversed-biased. Very neat!

Of all the sections, the power supply proved the most difficult to get working. Several chips were harmed while testing the circuit. There is nothing quite like that puff of “magic smoke” coming from a completed board. The MCP73831 battery charger tended to spectacularly blow whenever a battery or live USB jack was inserted/removed. To avoid it, I added capacitors (C8, C13) across the LIPO and USB jack inputs. Don’t leave out these caps.

I also managed to install the MOSFET backwards during PCB design, which was rather dangerous: the body diode was no longer reversed-biased when 5V was applied, and current flowed directly into the battery, bypassing the charger.

The STM32 microcontroller

I choose this particular micro because it is the same microcontroller in the Blue Pill. This makes it very easy to use the Arduino IDE. And it makes it easy to debug, using a breadboarded Blue Pill for comparison. You only need a handful of components: the micro itself, 4 bypass caps, and an oscillator. I included a reset circuit, which you don’t need. I also added a diagnostic LED/resistor combo, which is very handy and comforting during the build, but not needed for clock operation. I used the same circuit in one of my previous projects, the Pocket Tutor. Read that article for more information.

The Display

I cheated on the “built entirely from scratch” goal and used a 3.2” TFT display module instead of a naked display. It was easier to troubleshoot the PCB without being tethered to a display via a fragile, flexible cable. It was also easier to connect/disconnect a display on 0.1” header pins than one using a tiny 40-pin FRC connector.

I have purchased a few of these display modules from eBay, and have noticed differences in how the backlight pin is handled. Most boards employ an NPN transistor as a “low-side switch” between the LED cathodes and ground. The board’s backlight pin drives this circuit and requires only a few mA of current. However, some manufacturers, in an effort to save a fraction of a cent, omit the NPN entirely and tie all of the backlight LED anodes directly to the backlight pin. You can connect this pin to power, but the current draw is too high to drive from a microcontroller. So, I added an optional PNP high-side switch to the PCB. If your display has the NPN transistor, fit jumper JP1 instead of the PNP driver.

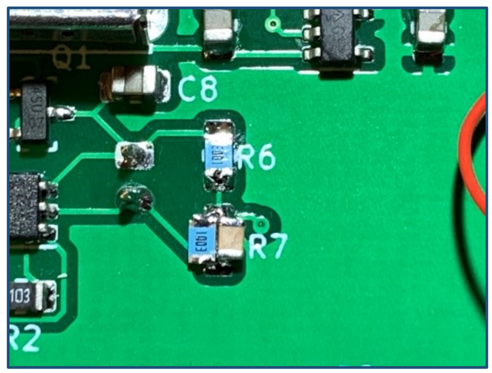
GPS Module

I like the uBlox NEO series, which is widely copied, so I used it for this circuit. It requires only 1 bypass cap, a handful of external components for an active antenna, and an optional LED/resistor. The Gotcha? To my surprise, the NEO-M8N module has suddenly become unavailable in the US. As of this writing, you must order it directly from uBlox and pay an exorbitant shipping fee. I did not see that coming, and it is the first time that I've not been able to obtain an active, commonly-used part. I anticipate that it will be restocked in the future.

Audio Interface

In the Morse Tutor and Pocket Tutor projects, I used PAM830x audio amplifiers for potentially loud outputs. But this clock would only need a small sound for alarms or chimes. My first choice was a piezo element, but I was disappointed in the sound quality. Instead, I used a small, flat oval-shaped speaker which adheres to the board with two-sided tape. These can be obtained on eBay for a few dollars. A single NPN transistor is sufficient to drive one. A snubber diode across the speaker would be helpful to reduce power line noise, but I didn't need it.

Battery Monitor



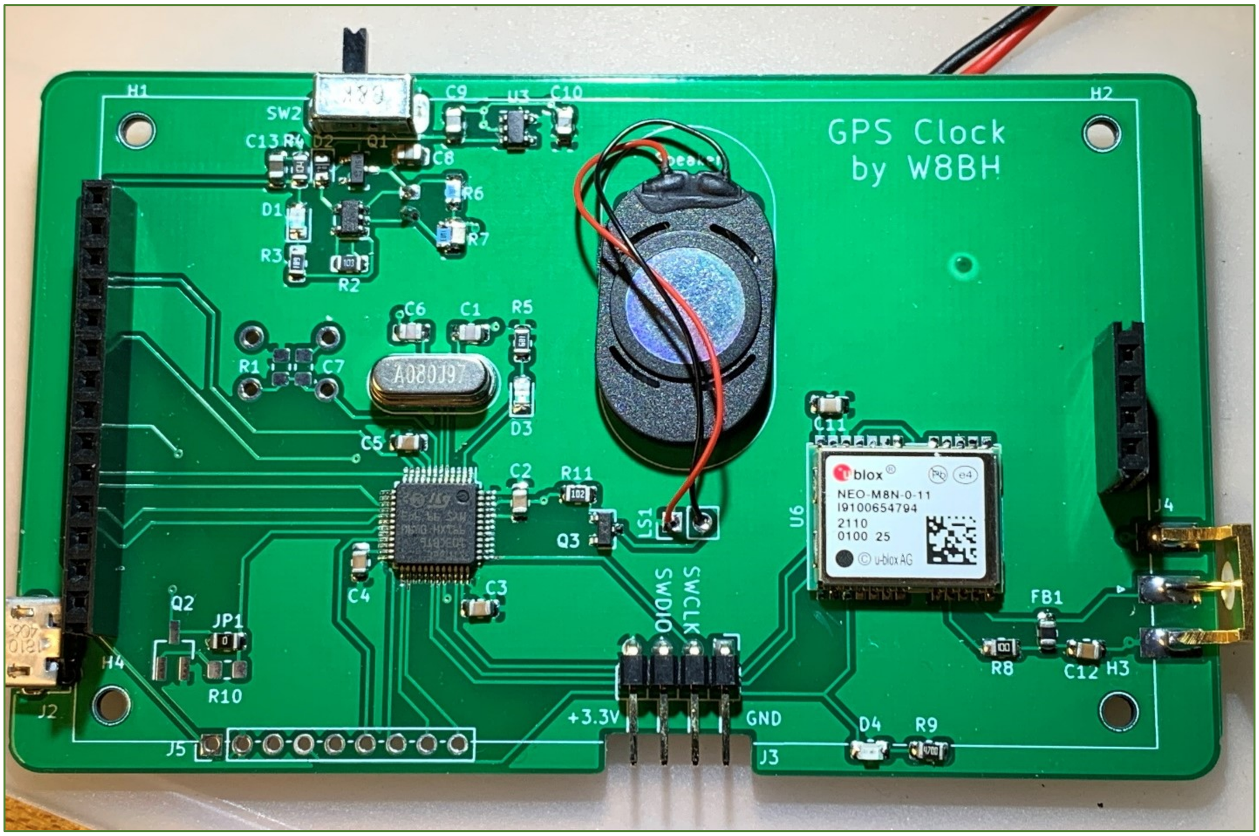
Two SMD components on single set of pads

Initially I opted for a very-low-current voltage divider for monitoring the LiPo battery. The battery voltage, which reaches 4.2V on a full-charge, exceeds the maximum allowable voltage on the microcontroller input. Dividing the voltage in half works, and I used this two-resistor divider without any problems on my pocket tutor. For reasons that I do not completely understand, the same circuit and same microcontroller did not work this time. I consistently obtained voltages that were 0.1 to 0.2 volts too low – making it unusable for battery monitoring. After almost giving up, I heeded the advice of a friend and searched the internet. In some microcontrollers, a 100K/100K divider does not provide enough current to adequately charge the internal sample-and-hold capacitor. I added a 100nF capacitor across the analog input, thereby providing a "current reservoir" for the internal cap. Rather than remake the circuit board, I opted to fit this extra cap side-by-side with the resistor, nudging R7 to the left of its pads (see photo).

I could have also reduced the values in the resistive-divider, but this would have also increased current draw on the battery when the unit is off, thereby shortening battery life.

The PCB

All of the components fit on a 60x100 mm board, when plenty of spare room. There is nothing especially critical about layout, except that the trace from the GPS receiver to the antenna jack should be short and reasonably straight. Bypass caps are kept close to the circuits they serve. Read these notes about the design of a similar project. The Gerber files for the PCB are available on GitHub.



Construction

Construction notes for the Pocket Tutor article generally apply to this board as well. I suggest fitting the micro-USB jack first, then adding the parts nearest to SW2 (C13, R4, D2, Q1, C8), and then soldering in SW2. Add the 3.3V regulator and its caps C9, C10. Confirm stable 3.3V power to the board. After that, install the STM32 microcontroller and associated parts. Confirm the STM32 is working by uploading a blink sketch via J3. Make sure that the micro-USB jack is working well and stable before soldering in the adjacent display header pins; it is difficult to remove or resolder the jack after the header is in place. Add the remainder of the battery charging circuit and audio circuit as you see fit. Since the GPS receiver module is the most expensive component on the board, save the GPS circuit installation for last.

73,

Bruce.

Last updated: 9/25/2021

Bill of Materials

	Value	Footprint	Qty
C1, C6	20pF	C_0805_2012Metric_Pad1.15x1.40mm_HandSolder	2
C12, C14, C2, C3, C4, C5	100nF	C_0805_2012Metric_Pad1.18x1.45mm_HandSolder	6
C7, C11	1uF	C_0805_2012Metric_Pad1.18x1.45mm_HandSolder	2
C13	4.7uF	C_0805_2012Metric_Pad1.15x1.40mm_HandSolder	1
C8, C9, C10	10uF	C_0805_2012Metric_Pad1.15x1.40mm_HandSolder	3
D1	Charging LED (blue)	LED_0805_2012Metric_Pad1.15x1.40mm_HandSolder	1
D2	D_Schottky	D_0805_2012Metric_Pad1.15x1.40mm_HandSolder	1
D3	Diagnostic LED (red)	D_0805_2012Metric_Pad1.15x1.40mm_HandSolder	1
D4	1PPS LED (yellow)	C_0805_2012Metric_Pad1.18x1.45mm_HandSolder	1
FB1	Ferrite Bead	C_0805_2012Metric_Pad1.18x1.45mm_HandSolder	1
J5	Conn_01x09_Male	PinHeader_1x09_P2.54mm_Vertical	1
J1	Lipo Battery Connector	JST_PH_S2B-PH-K_1x02_P2.00mm_Horizontal	1
J4	Molex 73251	SMA_Molex_73251-1153_EdgeMount_Horizontal	1
J3	Programming	PinHeader_1x04_P2.54mm_Horizontal	1
J2	USB_B_Micro	USB_Micro-B_Molex-105017-0001	1
JP1	0-ohm resistor	R_0805_2012Metric_Pad1.15x1.40mm_HandSolder	1
LS1	Speaker	PinHeader_1x02_P2.54mm_Vertical	1
Q1	DMP2045U-7	SOT-23-3	1
Q3	MMBT4401	SOT-23	1
Q2	MMBT4403	SOT-23_Handsoldering	1
R8	10	C_0805_2012Metric_Pad1.18x1.45mm_HandSolder	1
R9	470	C_0805_2012Metric_Pad1.18x1.45mm_HandSolder	1
R3, R5	680	R_0805_2012Metric_Pad1.15x1.40mm_HandSolder	2
R10, R11	1K	R_0805_2012Metric_Pad1.15x1.40mm_HandSolder	2
R1, R2	10K	R_0805_2012Metric_Pad1.15x1.40mm_HandSolder	2
R4, R6, R7	100K	R_0805_2012Metric_Pad1.15x1.40mm_HandSolder	3
SW1	SW_Push	SW_PUSH_6mm	1
SW2	SW_SPDT	SW_Slide_1P2T_CK_OS102011MA1QN1	1
U1	3.2” TFT DISPLAY	MSP3218 (ILI9341 SPI)	1
U3	AP2112K-3.3	SOT-23-5	1
U2	MCP73831-2-OT	SOT-23-5	1
U6	NEO-M9N	ublox_NEO	1
U5	STM32F103C8Tx	LQFP-48_7x7mm_P0.5mm	1
Y1	8 Mhz	Crystal_HC49-U_Vertical	1