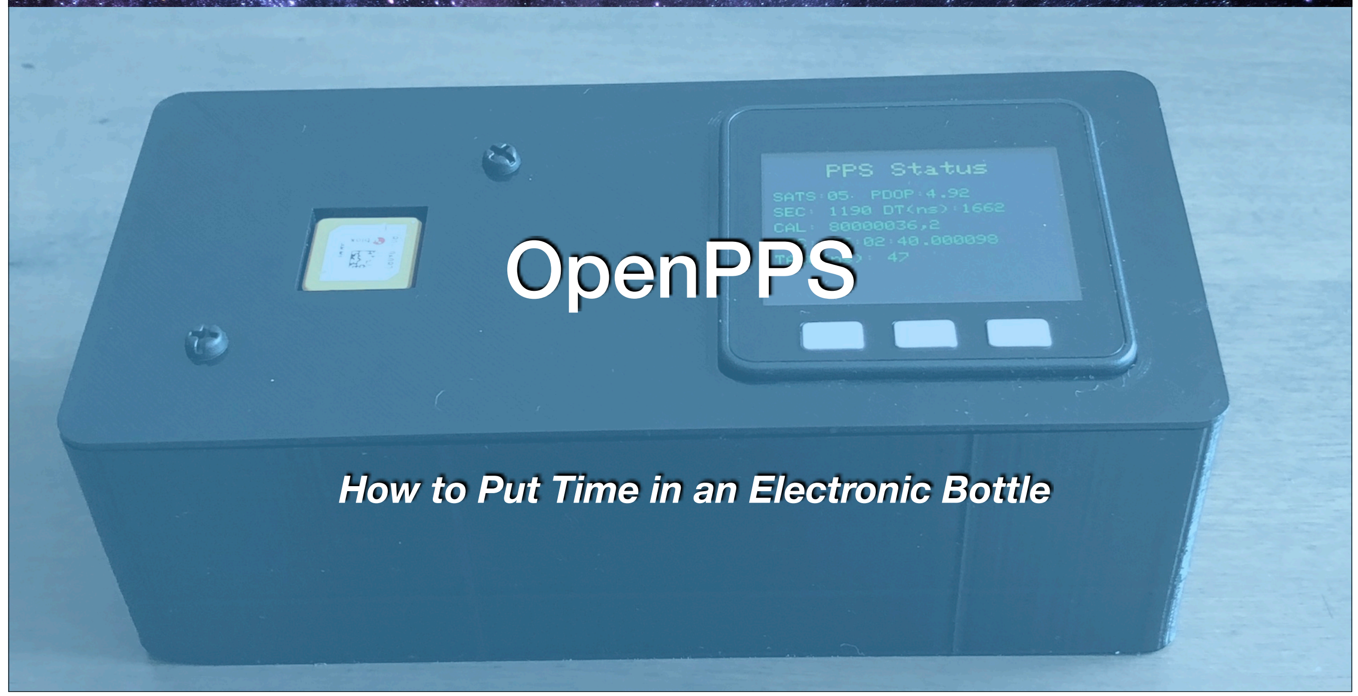
RocketManRC

Maker Faire

OpenPPS

How to Put Time in an Electronic Bottle



Presentation Objective

The title of this presentation was inspired by the late Jim Croce's song "Time in a Bottle"!

I am going to show how I created a precision time reference from an FPGA, stabilized oscillator, ublox GPS module and an ESP32.

The GPS only needs to be active occasionally therefore the time derived from the oscillator is saved in the "electronic bottle".

http://www.rocketmanrc.com

About Me

- I had a 40 year career in several industries including Aerospace a couple of times.
- My technical area of expertise is embedded systems, both hardware and software.
- I live in Atlantic Canada where we also have unpredictable weather like here!
- For the past year and a half I have been privileged to be a full time Maker, but I have always been an electronics hobbyist.
- I like to say that I am on “permanent sabbatical”!

GPS

- I've been working with GPS since 1994 starting with a Trimble development kit. My main concern at the time was how we were going to fit anything like that in the hand held product we were designing! How little I knew then…

- In those days the GPS antenna had to be outside with a clear view of the sky to get a position fix. For development you needed an antenna on the roof and a long and expensive cable.

- Today GPS receivers are everywhere to show you where you are, in your phone, your car and even your watch.

- When I say GPS I really mean GNSS (Global Navigation Satellite System) which includes GPS (US), GLONASS (Russia), Galileo (EU), Beidou (China) and some regional systems.

GPS Time

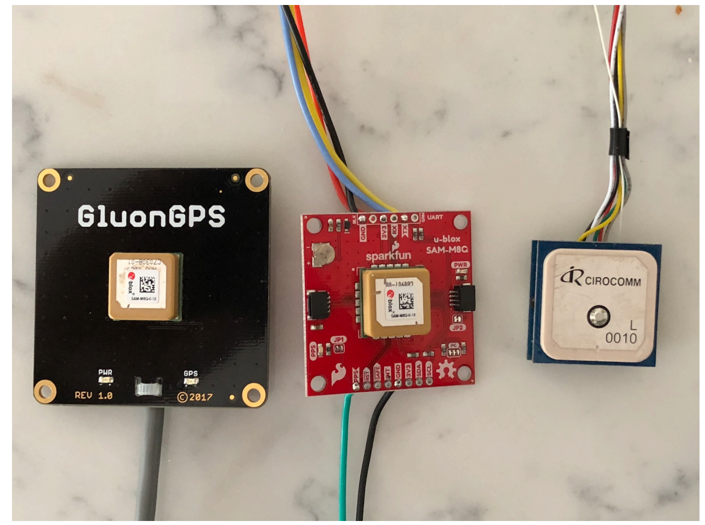
- Most people don't realize that GPS also gives time - very accurate and precise time - synchronized to atomic clocks in the GPS satellites.
- Coordinated Universal Time (UTC) is maintained by BIPM in Paris.
- National metrology institutes or observatories maintain physical realizations of UTC.
- The GPS satellites have atomic clocks that are steered to UTC by the operators.
- Every GPS receiver has an internal clock that is synced to the clocks in the satellites.
- An example of a commercial system is the Trimble Thunderbolt (requires an external antenna).

Trimble ThunderBolt Dev Kit

Frequency vs Time (just so we are on the same page)

- An oscillator produces a signal with a frequency (1 Hz = 1 cycle/second)
- If you count Hz with a counter you get an elapsed time. For example with a 10 MHz signal counting to 10,000,000 will give exactly one second.
- If you want time of day you have to synchronize to UTC.
- How often do you have to do this? It depends on the requirements and the accuracy and stability of the oscillator!

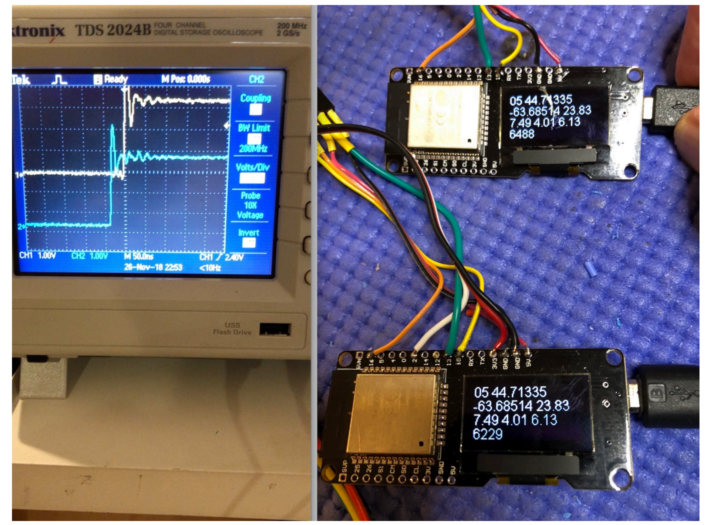
Some Current GPS Modules (ublox)



PPS (Pulse per Second)

- GPS receivers give pulses (PPS) that are 1 second long and within 10 to 50 ns of UTC.
- Usually PPS is connected to an LED to indicate the GPS has a fix and a position is available.
- Of the three modules shown in the previous slide only the Sparkfun one brings out PPS to a solder pad.
- It is not difficult to modify a receiver to get access to PPS

PPS from two GPS



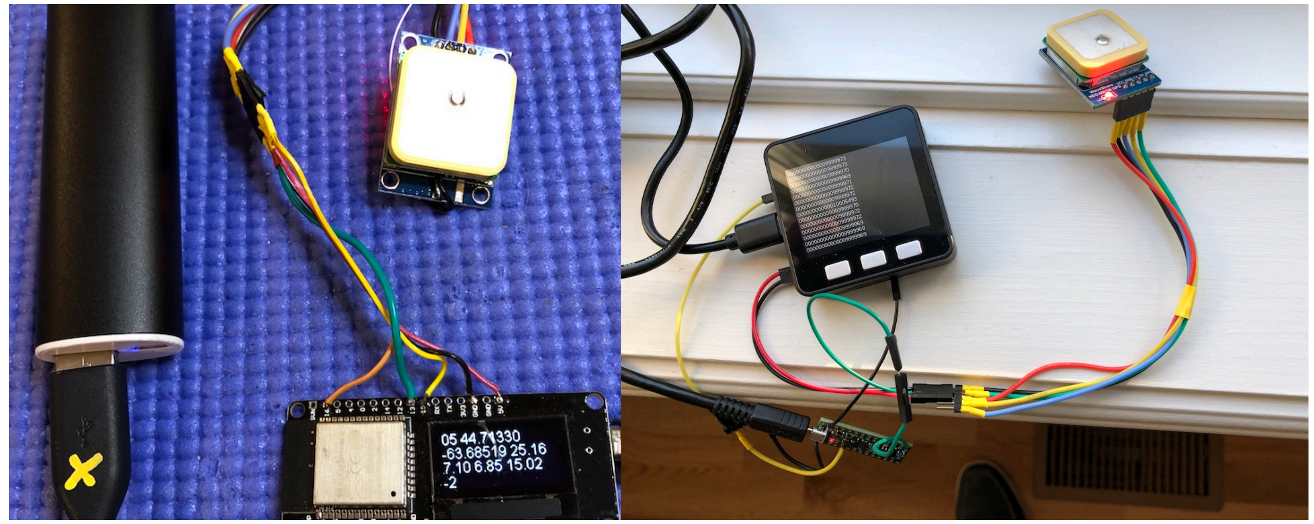
They stay within 50 ns of each other!

Modifying to get PPS Out

ESP32

- I thought I could do everything I wanted with an ESP32 which would be very convenient because it is easy to communicate with.
- I setup a 64 bit free running counter, generated an interrupt from PPS and read the counter.
- Trying to calibrate the counter was a problem because of lots of jitter and drift.
- I determined that this was from the ESP32's interrupt response time which was a lot longer than I expected and inconsistent.
- The best performance I could get was a drift in time of about 2 ms a day.

ESP32s with GPS

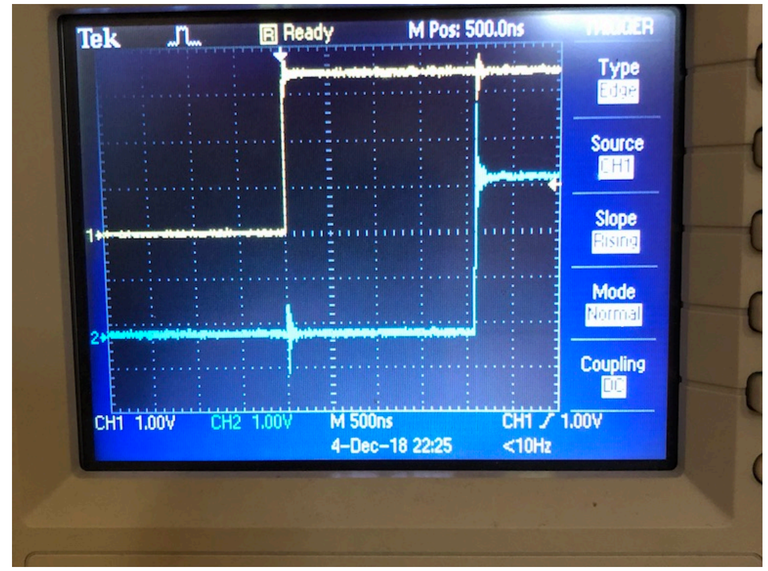


Note that I am using the M5Stack for data logging!

ESP32 Interrupt Response

- It turned out to be fairly well documented that the ESP32's interrupt response time is 2 to 3 us but I could see that it also wasn't consistent.
- I wanted to measure the response time and find the maximum.
- I used the standard technique of setting an output pin in the interrupt handler and comparing it to the PPS input with the 'scope.
- I could see the latency and jitter but not so easy to measure it!
- I ended up using an FPGA to do the measurement.

ESP32 Interrupt Response



500 ns/division = 2.2 us

Trying to Improve the ESP32 Performance

- I tried many different things with the ESP32 to improve performance including moving the interrupt handler to the system core and using the ESP-IDF instead of the Arduino framework.
- All of this had little effect and I wanted a better solution.
- I still wanted to retain the ESP32 because of its communications ability.
- I decided to move the counter logic to the FPGA. Down a big rabbit hole!
- I am going to talk about my further adventures with FPGAs this afternoon at 1:00 pm. It is also documented on my website in the OpenPPS project description.

Still Not Good Enough

- After using the FPGA for the counter logic the performance was better but still not what I was hoping for.
- I started looking into the performance of different types of oscillators that I could use with the FPGA (it can use any pin as a clock input).
- The accuracy and stability of oscillators is a big topic of its own.
- Down a another rabbit hole!

RocketManRC

Maker Faire

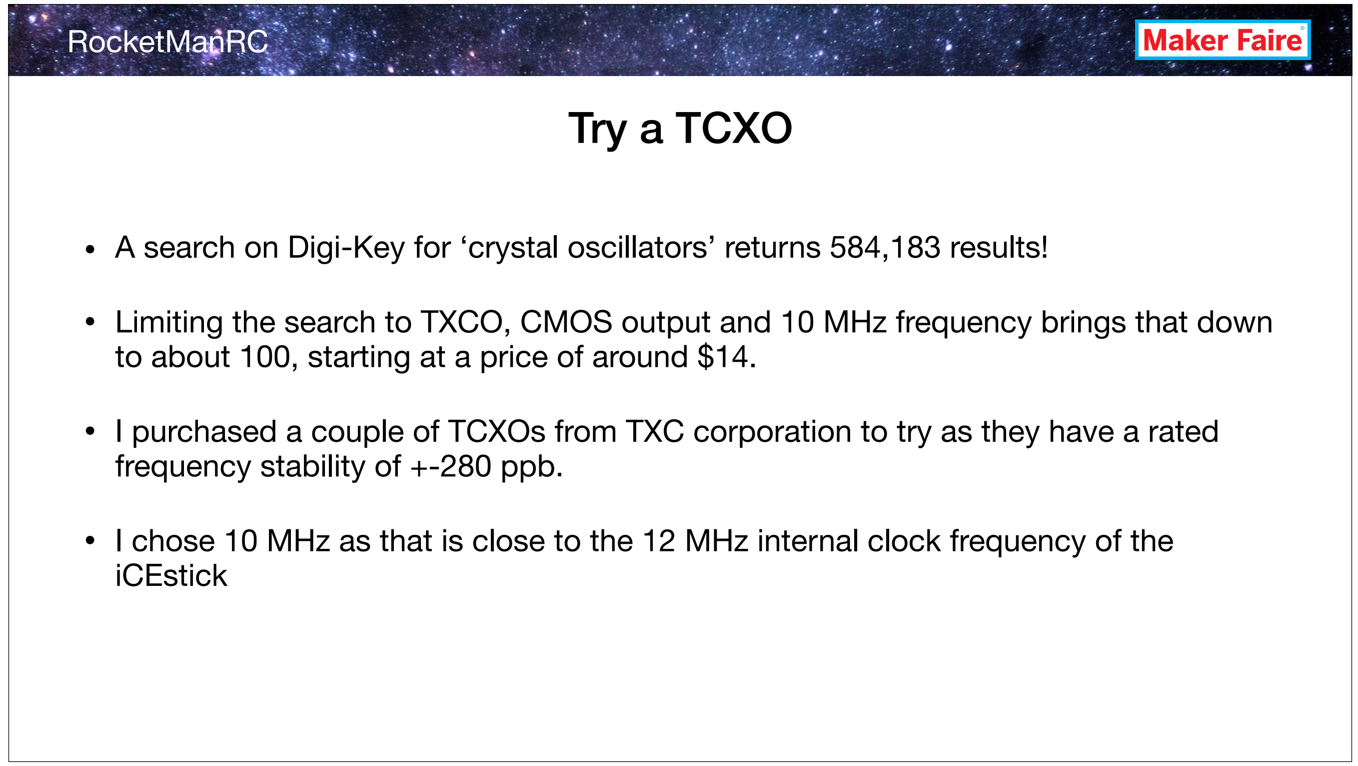
Oscillators

Oscillator Type	Frequency Stability (In severe temperature environments e.g. –40°C to +75 °C, and high slew rates)
Crystal Oscillator (XO)	$1x10^{-4}$ to $1x10^{-5}$
Temperature Compensated Crystal Oscillators (TCXO)	$1x10^{-6}$
Microcomputer Compensated Crystal Oscillators (MCXO)	$1x10^{-7}$ to $2x10^{-8}$
Oven Controlled Crystal Oscillators (OCXO)	$1x10^{-8}$ to $3x10^{-10}$
Poor Man's Rubidium ??? **High-Precision Double Oven Crystal Oscillator (DOCXO)**	**$1x10^{-10}$**
Rubidium Atomic Frequency Standards (Rb) [–10°C to +60 °C]	$3x10^{-10}$ to $7x10^{-11}$
Cesium Atomic Standard (Cs) [0°C to +50 °C]	$3x10^{-11}$ to $3x10^{-12}$

A good computer clock is stable to 100 ppm which is 5 minutes / month

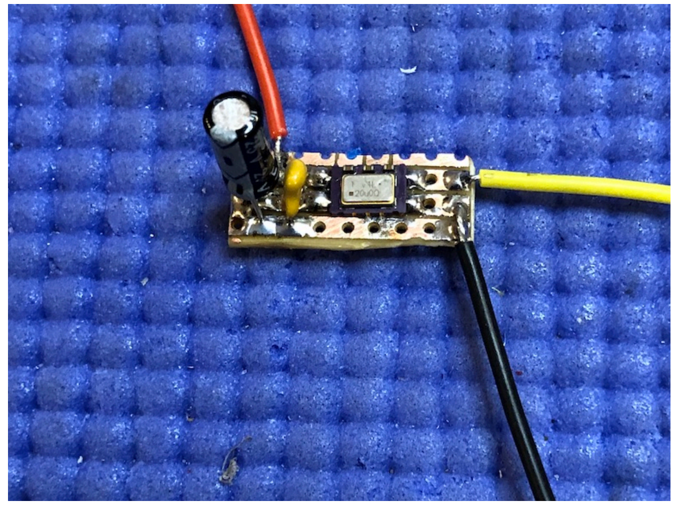
The best atomic clock is stable to 2.3 x 10-16 = 1 second in 138 million years

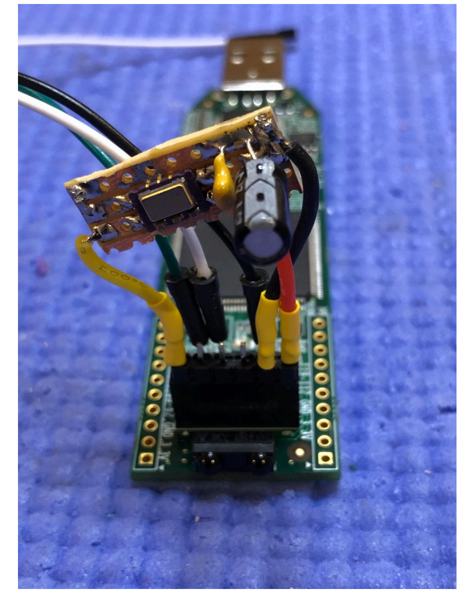
GPS system maintains a time transfer accuracy of less than 40 nanoseconds relative to UTC, 95% of the time.



TCXOs are readily available and not too expensive ($10 to $50)

Trying a TCXO



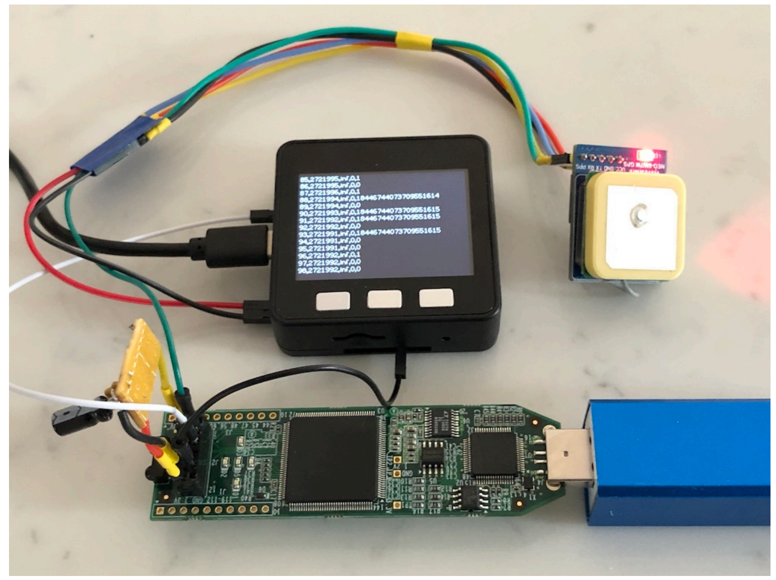


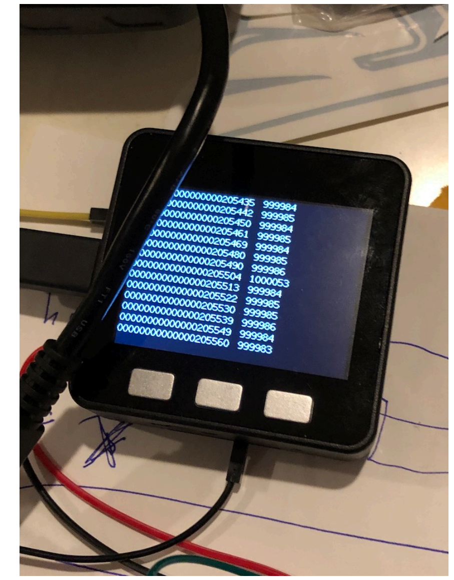
Temperature Compensated Crystal Oscillator

RocketManRC

Maker Faire

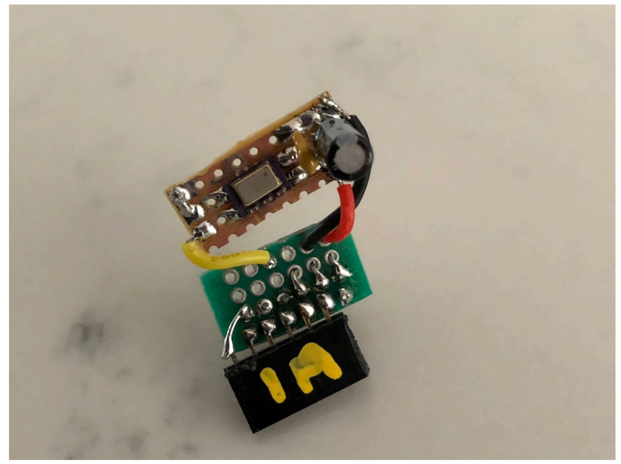
M5Stack used for logging test data

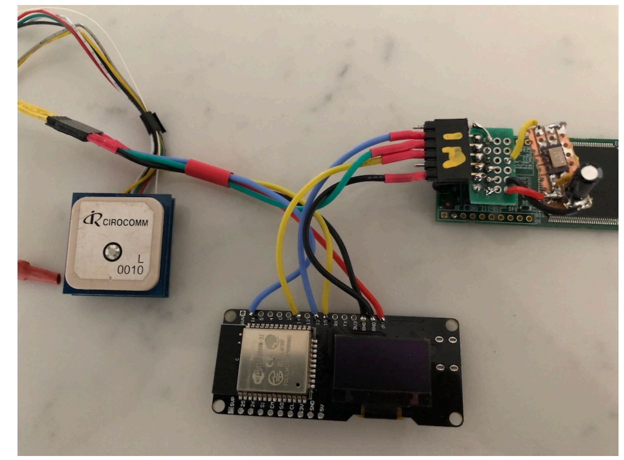




M5Stack is a self contained ESP32 based computer with TFT screen, buttons and SD Card

Wiring Harness for FPGA PMOD Connector



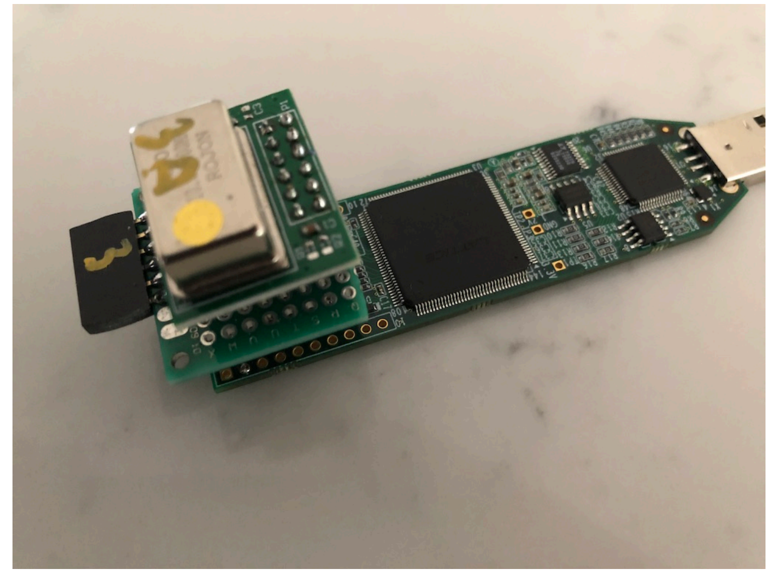


Created a standard wiring harness to be able to compare oscillator performance

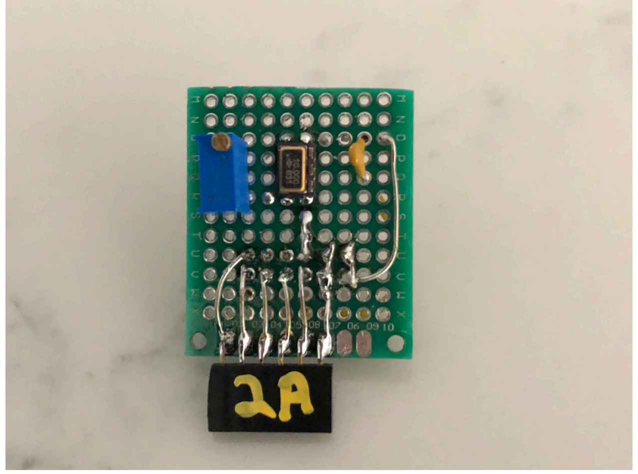
RocketManRC

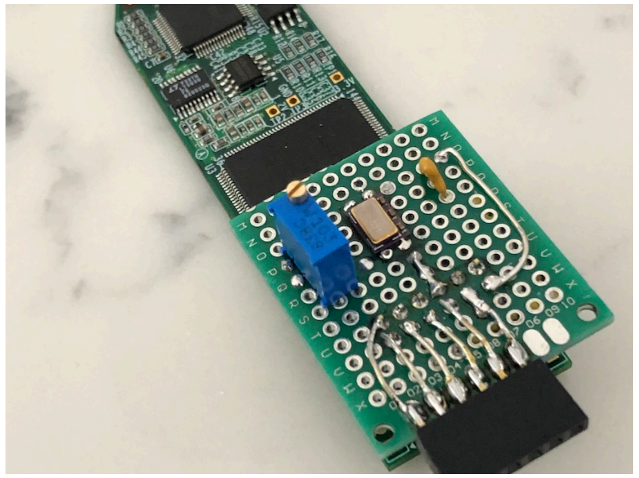
Maker Faire

$5 TCXO from Aliexpress



VCTCXOs

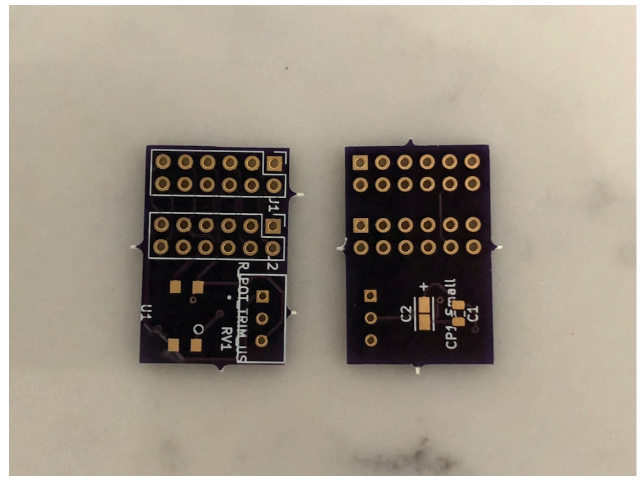


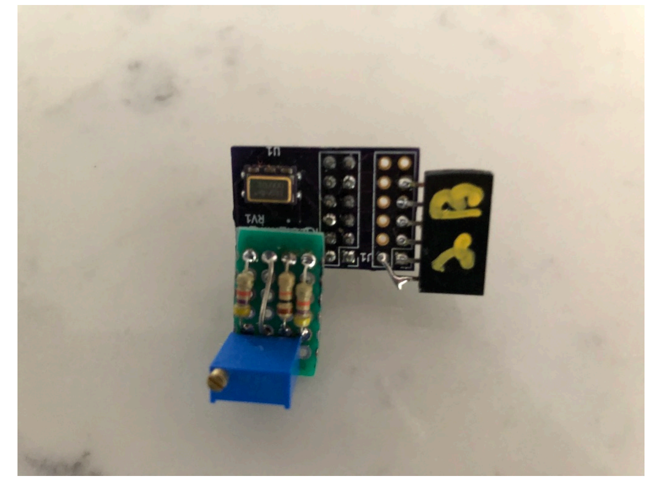


VC = voltage controlled

RocketManRC Maker Faire

Printed Circuit Board for PMOD Connector

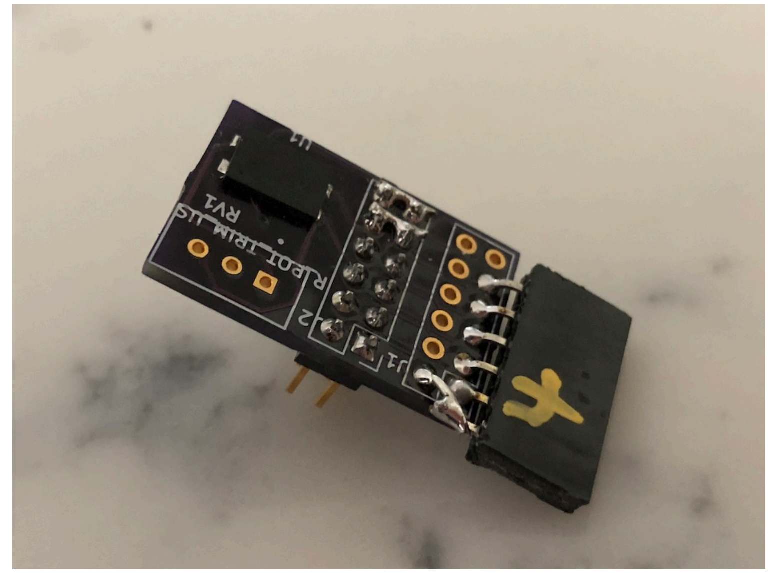




It was getting tedious to hand build these oscillator boards so had to lear KiCAD and design a board.

Resistors added to reduce “tuning range” of the trim pot.

MEMS Technology TCXO

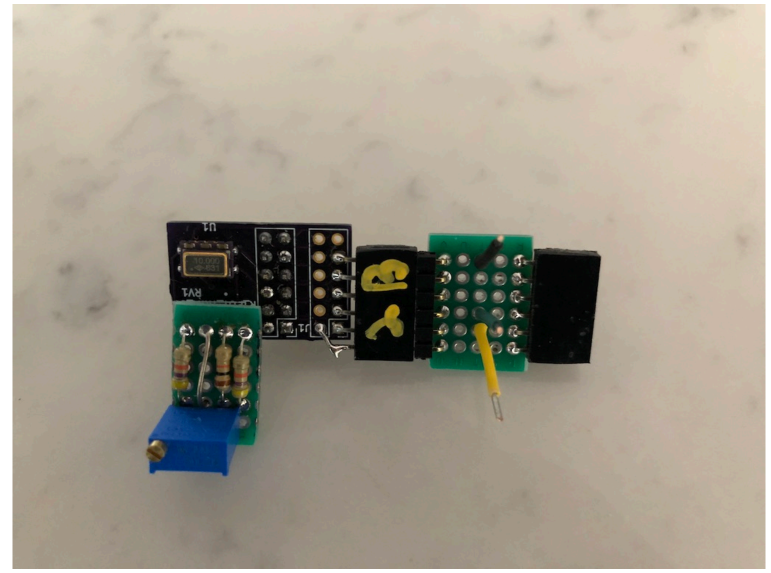


This does not use a crystal so aging problems should be reduced. Not sure yet.

RocketManRC

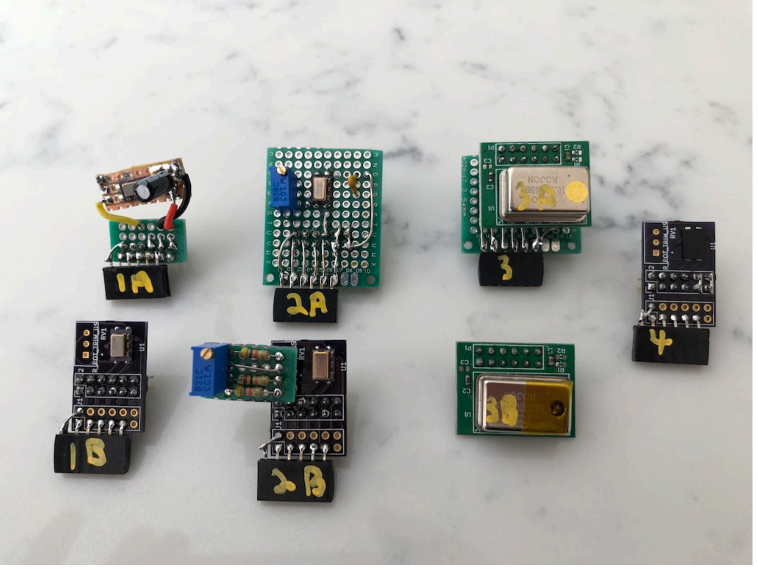
Maker Faire

“Extender” Board



To get the scope probes on.

My Collection So Far…

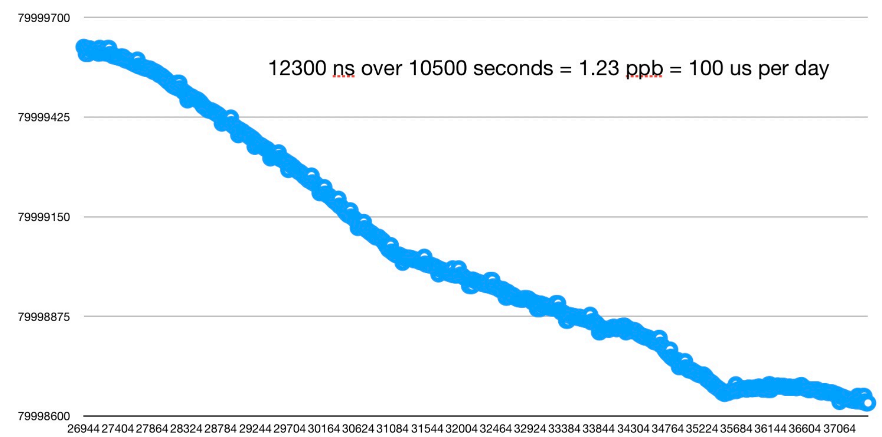


Results?

- 20 times less time drift than using just the ESP32.
- I haven’t been able to do enough testing to pick a winner among the ones I tested but the VCTCXOs don’t seem as stable (probably because of the trim pot).
- The aging is evident in all of these oscillator and can probably be corrected for.

RocketManRC

Typical drift with TCXO

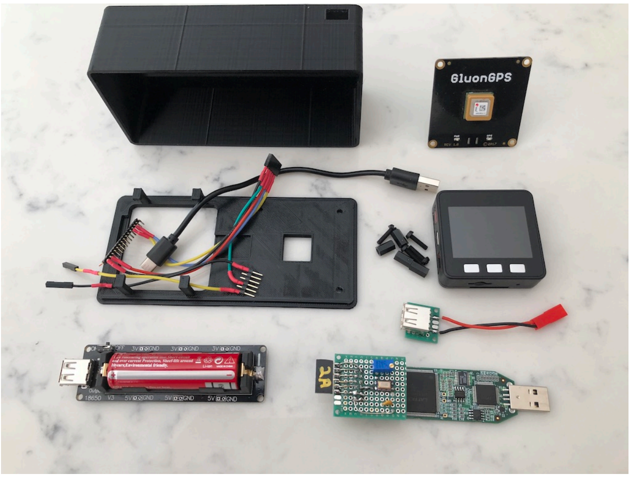


In theory this should be 100,000 times better stability that the crystal but in practice the crystal oscillator was better than spec.

RocketManRC

Maker Faire

PPS in a Box!



Needed to put it all in a box for the Maker Faire!

RocketManRC

Maker Faire

PPS in a Box!

This power supply with the 18650 LiON cell change power and charge at the same time.

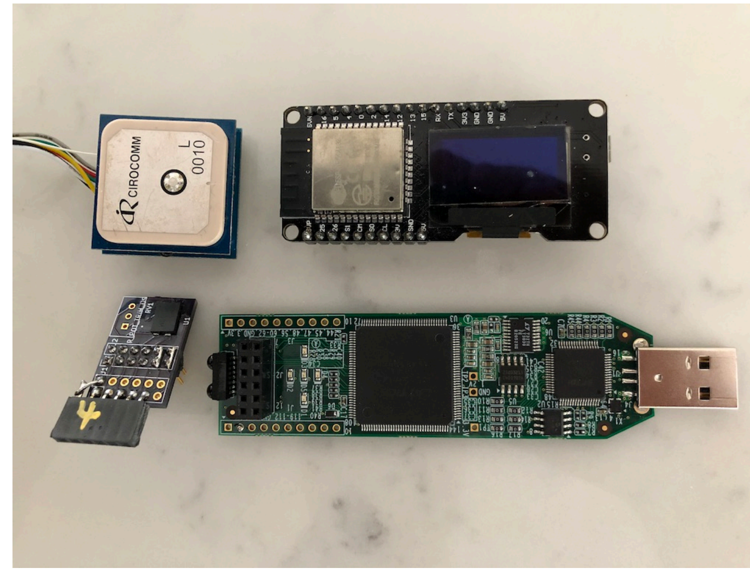
It still works!

It doesn't always work after it is in the box!

Lessons Learned

- An ESP32 + FPGA is a powerful combination. I wish there was a single board that had both on it.
- Precision timing requires stabilized oscillators.
- ublox M8 GPS receivers have amazing time performance when setup to be optimized for time applications
- Not all ESP32 boards are equal. I really like the M5Stack for development and prototyping.
- I chose to use KiCAD to design my oscillator board for the PMOD connector. A steep learning curve!

Lessons Learned



M8 GPS + software Library with enhancements for time performance.
TCXO and VCTCXO oscillators + circuit board with KiCAD
iCE40 FPGAs
ESP32 interrupt handling performance

Future Work

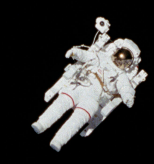
- My PPS in a box doesn't really do anything yet except test oscillators!
- Put a NTP server on the ESP32. It would drift only about 3 ms per month.
- Implement automatic calibration and also try aging prediction and compensation.
- Implement a secondary level of temperature compensation with a temperature sensor and code in the ESP32.
- Try the Ebay OCXO.
- Try the Connor Winfield VCTCXO and VCOCXO.
- Test the performance the S3231 RTC after tuning the aging registers.
- Get a used FE-5680A Rubidium Atomic Frequency Standard off of Ebay?

Final Comments

“The Journey is the reward”

— Steve Jobs

Thanks to Digi-Key for sponsoring this stage!



Questions?