Complete Detailed Design: Phase-End Summary

Individual and Team Vision for Complete Detailed Design Phase:

Team plan for this phase:

We planned to finish the majority of our detailed design for the space time card within this phase. This includes revisiting the past materials developed within the preliminary design phase. In order to allow for easy troubleshooting in the future a breakout board for the MicroZed 7010 SoM was planned to be developed. Similarly, the layout for the final timecard was planned. The digital phase lock loop (DPPL) was to be furnished on MATLAB (Simulink), with the addition of environmental characteristics, clock aging, and supply stability. The overall 3d model of the timecard was to be furnished if possible. Future test plans for the time card were furnished through the use of the Raspberry Pi GNSS development board. Within any spare time the Open Compute Time Card design was to be analyzed further in order to reuse or repurpose their design elements.

Accomplished during this phase:

The development breakout board for the MicroZed 7010 SoM was developed. The overall timecard PCB design was developed with a few placeholder components pending a functional prototype. Upon discussion, both existing and future designs should be reviewed by a qualified individual (Professor Carlos Barrios). The Raspberry Pi GNSS testbench was partially assembled, due to MSD ordering issues. The MATLAB (Simulink) DPPL simulations encompassed the main erroneous characteristics. The 3d models of the current state of the timecard were generated. Overall, a lot of this progress is “living” and will be updated within the next phase and near future.

Feasibility [Analysis, Simulations]:

Feasibility Simulation - Hold Over Drift Specifications:

Preface

This document calculates, compares, and contextualizes the holdover specifications of the SA-45 CSAC and the SiT5356 TCXO, along with the maximum amount we would be able to improve these specifications through steering algorithms that target environmental factors and aging.

Time error of CSAC (SA-45)

$$Time\ Error = E(t) = E_0 + y_0 t + \int_0^t \int_0^t a dt + \int_0^t y_e(t) dt + [t \times \sigma(\tau = t)] \quad [1]$$

$E_0 = Initial\ time\ error$

$y_0 = Initial\ frequency\ error < 10^{-12}\left[\frac{Hz}{Hz}\right]$ [2]

$a = Aging\ rate\ (frequency) < \frac{9*10^{-10}\left[\frac{Hz}{Hz*30Days}\right]}{86400*30[\frac{s}{30days}]} = 3.472 * 10^{-16}\left[\frac{Hz}{Hz*s}\right]$ [2]

$y_e(t) = Frequency\ error\ due\ to\ environmental\ conditions < 5 * 10^{-10}$ [2]

$\sigma(\tau = 86400s) = Allen\ deviation = 10^{-11}$ [3] Page 36

$Assume\ linear\ frequency\ progression\ due\ to\ each\ frequency\ drift\ factor.\ This\ is\ the\ absolute\ maximum\ drift.$

$E_{CSAC,max}(t = 24hours = 86400s) = 0 + \left(10^{-12}\right)(86400s) + \frac{1}{2}\left(3.472 * 10^{-16}\right)(86400s)^2$

$+ \frac{1}{2}\left(5 \times 10^{-10}\right)(86400s) + \left[(86400s)\left(10^{-11}\right)\right]$

$E_{CSAC,max}(t = 86400s) = 8.64 * 10^{-8}s + 1.296 * 10^{-6}s + 2.16 * 10^{-5}s + 8.64 * 10^{-7}s = 2.38 * 10^{-5}s = 23.8\mu s$

$The\ best\ that\ we\ can\ correct\ for\ these\ errors\ is\ if\ we\ can\ fully\ correlate\ the\ aging\ and\ the\ environmental\ error\ over\ time\ and\ temperature\ and\ correct\ while\ in\ holdover\ accordingly.$

$E_{CSAC,best\ corrected,max}(t = 86400s) = 8.64 * 10^{-8}s + 8.64 * 10^{-7}s = 9.504 * 10^{-7}s = 950ns$

$Assume\ absolute\ max\ means\ within\ 4\ standard\ deviations\ of\ mean\ (99.9937\%\ likelihood\ to\ be\ at\ the\ "max"\ or\ better)$

$E_{CSAC}(t = 86400s) =$

$\left[E_{y_0} \sim N\left(0, 2.16 \times 10^{-8}\right)\right] + \left[E_a \sim N\left(0, 3.24 \times 10^{-7}\right)\right] + \left[E_{y_e} \sim N\left(0, 1.08 \times 10^{-5}\right)\right] + \left[E_{\sigma(\tau)} \sim N\left(0, 2.16 \times 10^{-7}\right)\right]$

$E_{CSAC}(t = 86400s) \sim N\left(0, 1.06 * 10^{-5}\right)$ (from stochastic simulation)

$E_{CSAC,corrected}(t = 86400s) = \left[E_{y_0} \sim N\left(0, 2.16 \times 10^{-8}\right)\right] + \left[E_{\sigma(\tau)} \sim N\left(0, 2.16 \times 10^{-7}\right)\right]$

$E_{CSAC,corrected}(t = 86400s) \sim N\left(0, 2.19 * 10^{-7}\right)$ (from stochastic simulation)

Maximum Fractional Error per day:

$e_{CSAC,max} = \frac{4.54*10^{-5}s}{86400s} = 5.25 * 10^{-10} = 0.525ppb$

$e_{CSAC,corrected,max} = \frac{9.504*10^{-7}s}{86400s} = 1.1 * 10^{-11} = 0.011ppb$

TCXO (SiT5356)

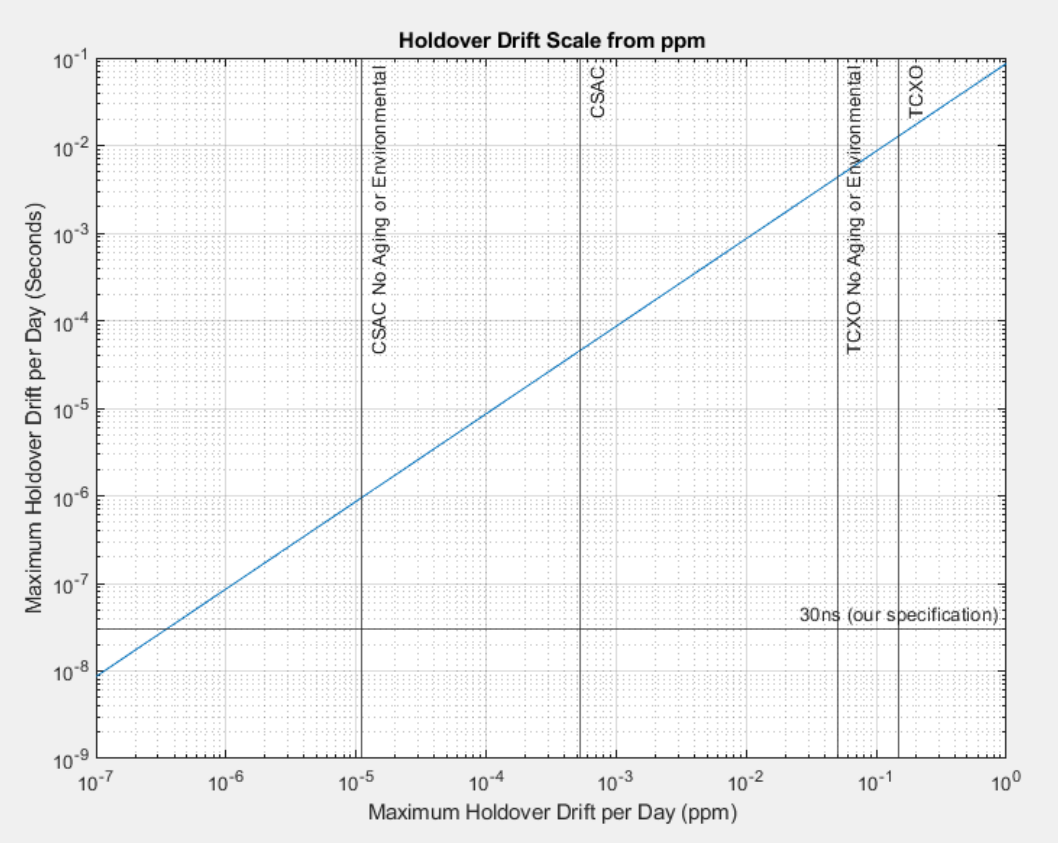
From the datasheet, the holdover specification is given as a max drift of 0.15ppm per Earth day in frequency
Maximum frequency stability over temperature is 0.1ppm per Earth day [4]

$E_{TCXO,max}(t = 86400s) = \frac{1}{2}\left(\frac{0.15}{10^6}\right)(86400s) = 0.00648s = 6.48ms$

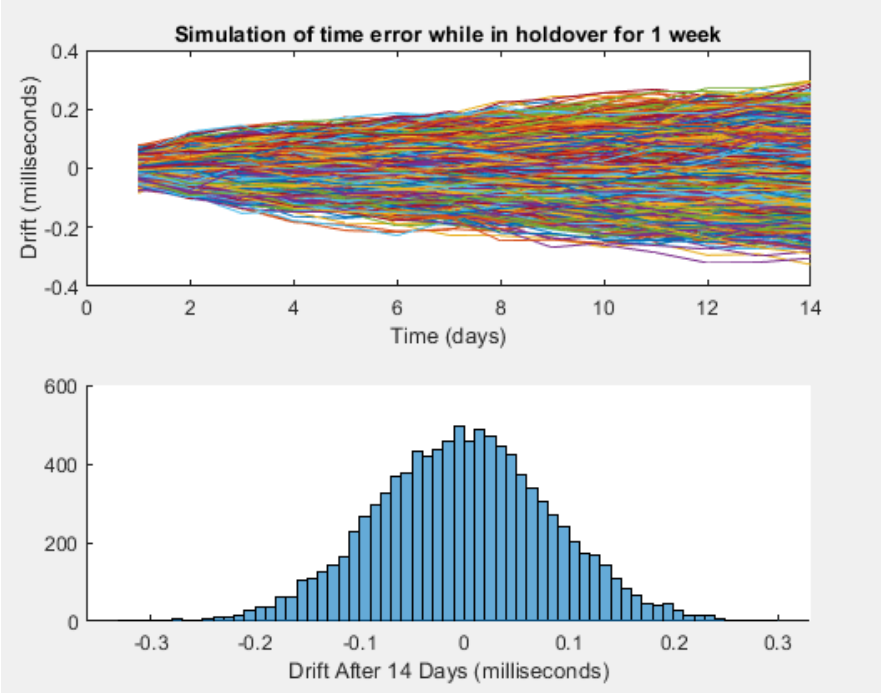
$If\ we\ can\ fully\ correct\ for\ environmental\ factors,\ we\ remove\ the\ 0.1ppm\ stability\ over\ temperature.\ Max\ aging\ rate\ is\ 2ppb\ per\ day,\ which\ is\ negligible\ comparatively.$

$E_{TCXO,best\ correction,,max}(t = 86400s) = \frac{1}{2}\left(\frac{0.15}{10^6}\right)(86400s) - \frac{1}{2}\left(\frac{0.1}{10^6}\right)(86400s) = 0.00216s = 2.16ms$

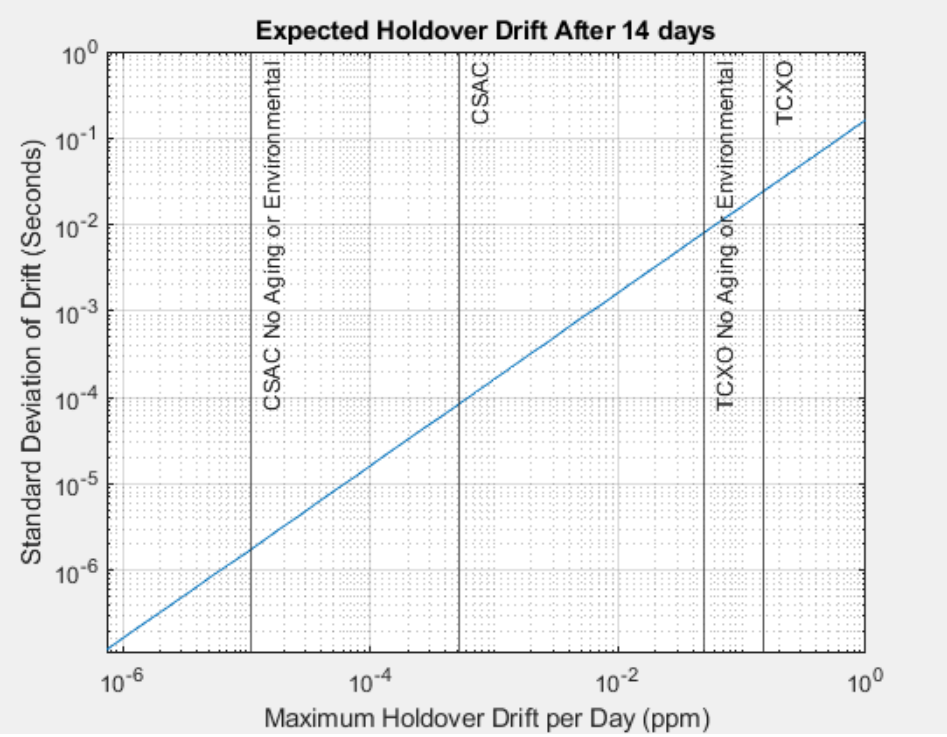
$e_{TCXO,max} = 0.15ppm \quad e_{TCXO,best\ corrected,,max} = 0.05ppm$



Conversion between ppm and seconds when referring to holdover drift specifications. Represents the maximum expected time error, in seconds, of an oscillator with a given ppm value.



Example stochastic simulation of 10,000 holdover time drift scenarios assuming normal distribution probability of drift with $\sigma = \frac{max\ error}{4}$. The figure shown is specifically using the non-corrected CSAC holdover specification (0.525ppb).



Standard Deviation of Time Error while in Holdover for 14 Days (from stochastic simulation). Assuming a normal distribution of time error each day with $\sigma = \frac{max\ error}{4}$.

[1] Vectron Holdover Oscillators Application Note https://ww1.microchip.com/downloads/aemDocuments/documents/VOP/ApplicationNotes/ApplicationNotes/Holdover+Oscillators.pdf

[2] SA-45 Datasheet https://ww1.microchip.com/downloads/en/DeviceDoc/00002985.pdf

[3] SA-45 Users Guide https://www.mouser.com/datasheet/2/523/Microsemi_CSAC_UserGuide_098-00055-000_D1-1114611.pdf?srsltid=AfmBOopsJ31mfG9VY86vBPsdfTfKj3h5XFMRbC7FOcGew0u8a-31Q39V

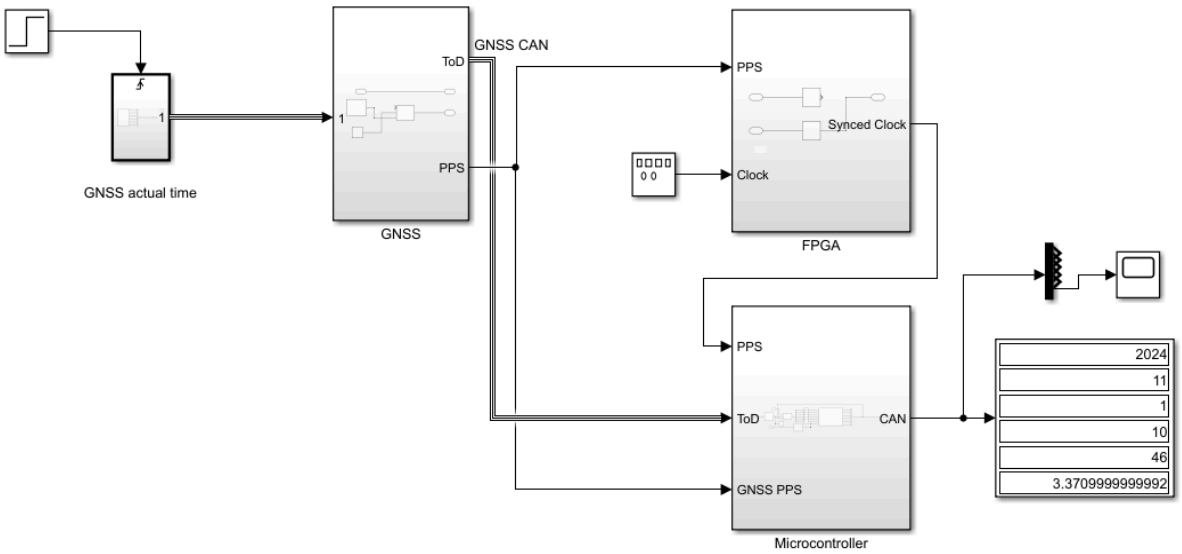
[4] SiT5356 Datasheet https://www.sitime.com/support/resource-library/datasheets/sit5356-datasheet

Feasibility Simulation - CubeSat Timecard Functional Emulation:

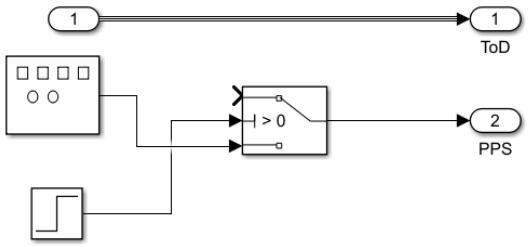
The MATLAB project can be found on the GitHub attached:

https://github.com/les4675/CubeSatTimeCard/tree/main/High%20Level%20Model

The goal is to create a model that emulates the Cubesat Timecard. The model has 3 major parts to it, the FPGA, Microcontroller, and GNSS. These parts are not fully flushed-out models; instead, they are quick approximations of what is happening in each system.

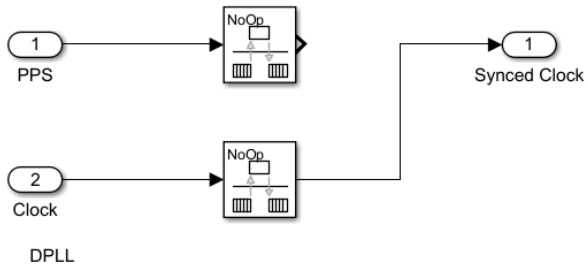


The model takes a GNSS input signal. This signal has two pieces of information that are necessary for what the time card is trying to do, the Pulse per second (PPS), and Time of Day (ToD). The function on the right grabs the time of day that is currently happening. On the left is the PPS signal that is being received.

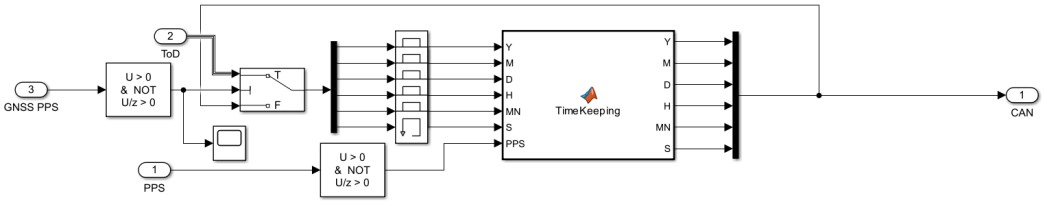



```
function [Y, M, D, H, MN, S] = fcn()
   eml.extrinsic('now');
   eml.extrinsic('datevec');
   Y = 0;
   M = 0;
   D = 0;
   H = 0;
   MN = 0;
   S = 0;
   [Y, M, D, H, MN, S] = datevec(now);
end
```

Once the signal is received, the PPS and ToD are used until disconnected. The FPGA then takes the PPS from the GNSS and syncs the clock that is onboard with that PPS in a Digital Phase Lock Loop (DPLL). This is done with the following system on the FPGA.



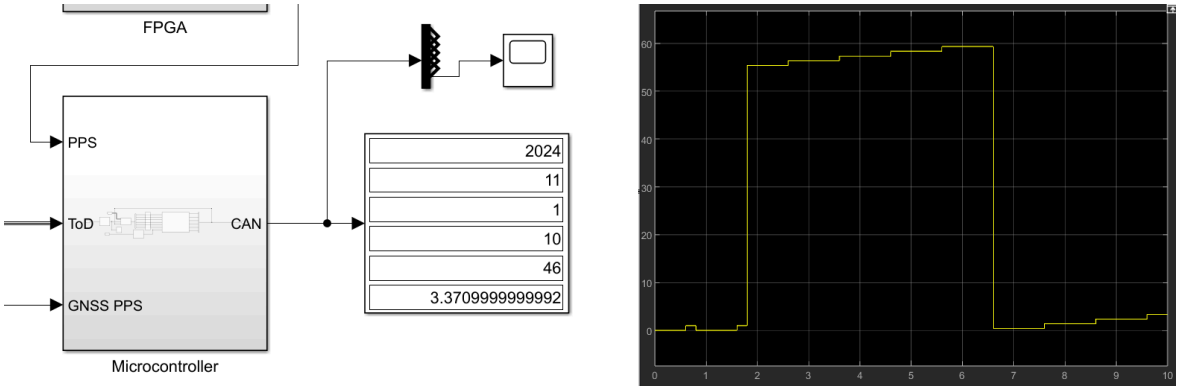
This synchronization clock is then used by the microcontroller to manage the time that is being kept. The signal can also be sent back out as an output for any other devices that want a PPS. The Microcontroller takes the ToD from the GNSS and saves it into memory. It will also take in a PPS to manage the time, incrementing the seconds based on the rising edge of the PPS.



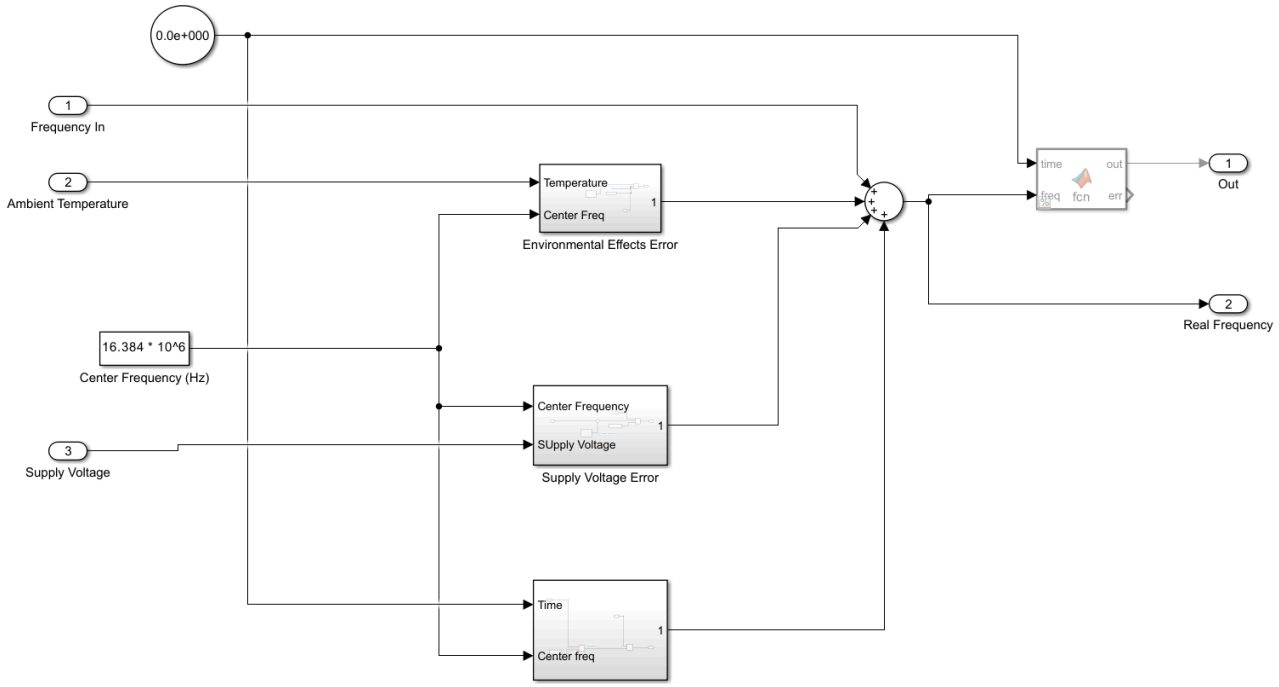
Here is the function that manages the timekeeping:

```
function [Y, M, D, H, MN, S] = TimeKeeping(Y, M, D, H, MN, S, PPS)

    if (PPS == 1)
        S = S + 1;
    end
    if (S > 60)
        S = S- 60;
        MN = MN+1;
    end
    if (MN > 60)
        MN = MN - 60;
        H = H + 1;
    end
    if (H > 24)
        H = H - 24;
        D = D + 1;
    end
    if (D > 30)
        D = D - 30;
        M = M + 1;
    end
    if (M > 12)
        M = M - 12;
        Y = Y+1;
    end
end
```

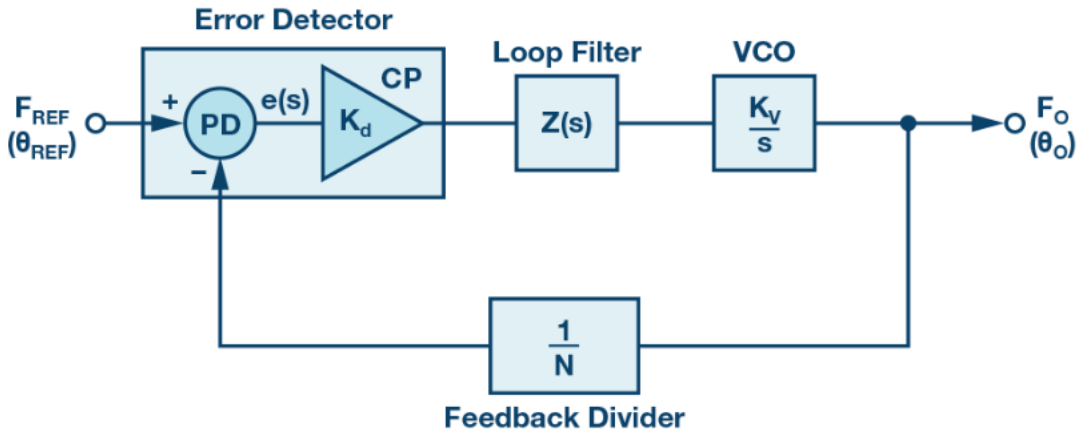
For simplicity, the assumption is made that a month is 30 days. When implemented in the future, there will be a lookup table to determine the number of days in each month. The following is a demonstration of the time being managed, handling seconds and minutes.



Over 10 seconds the system gets a GNSS signal in the first 2 seconds and then the ToD is stored in the main memory (MM). The system continues the time by incrementing the time on each rising edge. When the minute threshold is reached, it returns seconds to 0 and increments the minute.

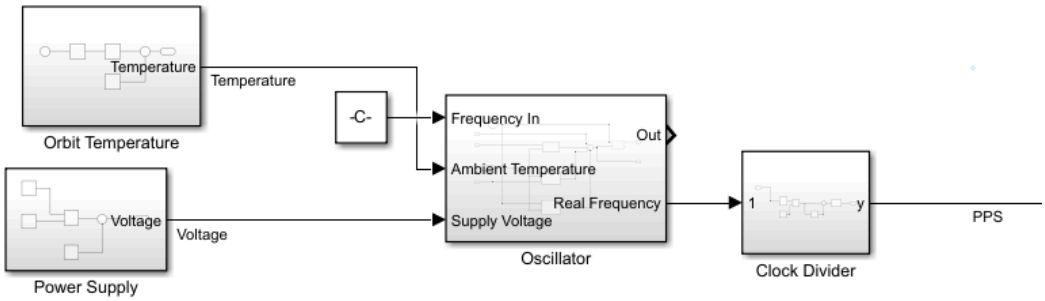


A realistic oscillator submodule based on the SA-45 was created to simulate both the deterministic errors of the clock and its steering capabilities. When being implemented in the simulation, the 16MHz signal had to be disabled due to the high sampling time and therefore simulation total time required. Instead, a second output is created for the real frequency of that signal, and this value is integrated outside the clock to create a PPS signal.

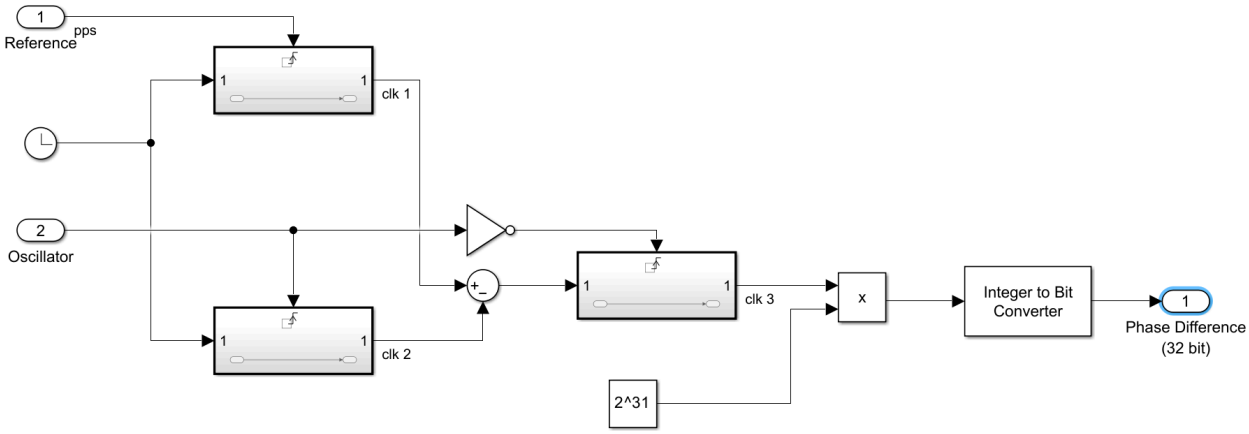


https://www.analog.com/en/resources/analog-dialogue/articles/phase-locked-loop-pll-fundamentals.html

This is an example diagram of a phase-locked-loop (PLL) circuit. This is what needs to be achieved in simulation, then in implementation, in order to lock the on-board oscillator to GNSS.



Profiles for the orbit temperature and the power supply were created to add deterministic errors to the system. The clock divider integrates the “real frequency” value and creates a pulse per second signal from that.



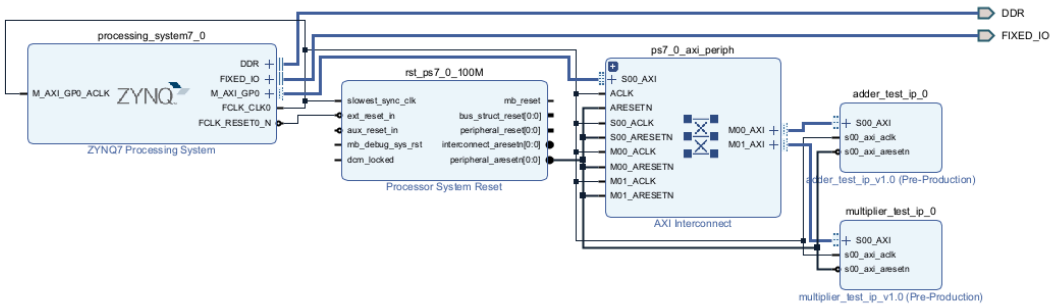
A block for a digital phase discriminator, a key component to the phase-locked-loop (PLL) was created in simulink. This takes in two PPS signals and outputs the phase difference between them as a signed 32 bit number that can be filtered by the loop filter.

Feasibility [Prototyping]:

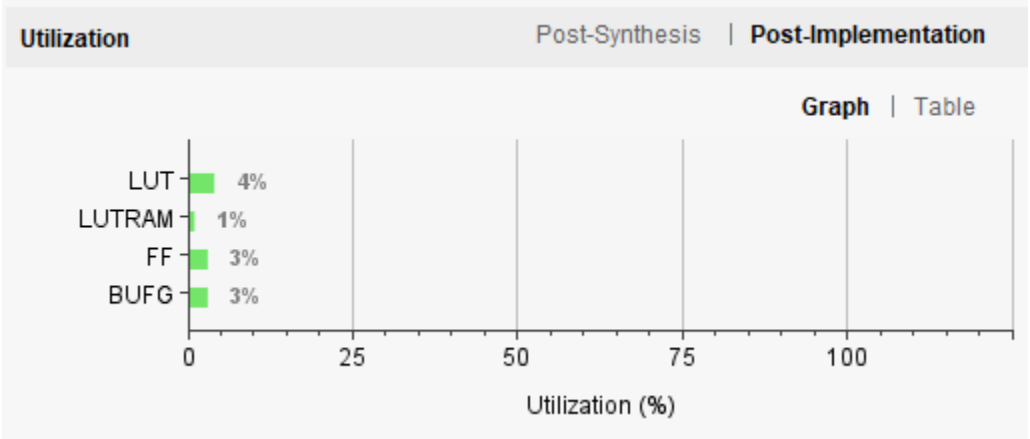
Creating the Embedded Environment:

One of the most important aspects of this project is developing the digital system and integrating it within our final deliverable PCB. We decided to implement our solution using a System on Chip (SoC). An SoC has Programmable Logic (FPGA) and a Processing System (ARM Core). The benefit of having both on a single chip offers an intuitive solution to avoid high frequency signals interfaced on a PCB between an isolated microcontroller and FPGA. Due to the complexity of this chip, the environment is a very difficult setup. Alongside product development, our team focused on creating an intuitive embedded environment so that once the design was complete as a Simulink/MATLAB concept it would easily be ported into reality using the FPGA and ARM Core of the SoC.

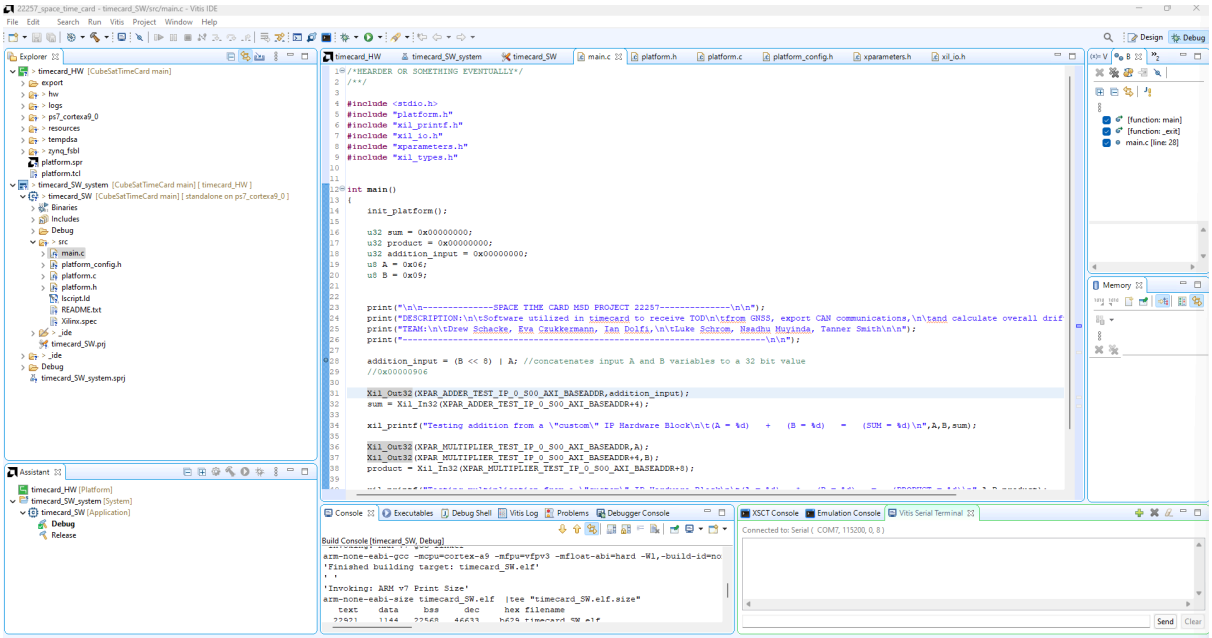
The first step was to create the hardware, which consists of a Zynq Processing System, interconnect, reset, and two custom hardware blocks that currently simulate placeholders for final logic. To verify the final environment placeholders were created to add and multiply to prove software communication with future custom hardware blocks.



The current environment barely scratches the surface of the SoC utilization!



After the hardware is created, it is exported from Vivado (Digital Design Software) and imported into Vitis (Embedded Design Software) which appears in the image below.

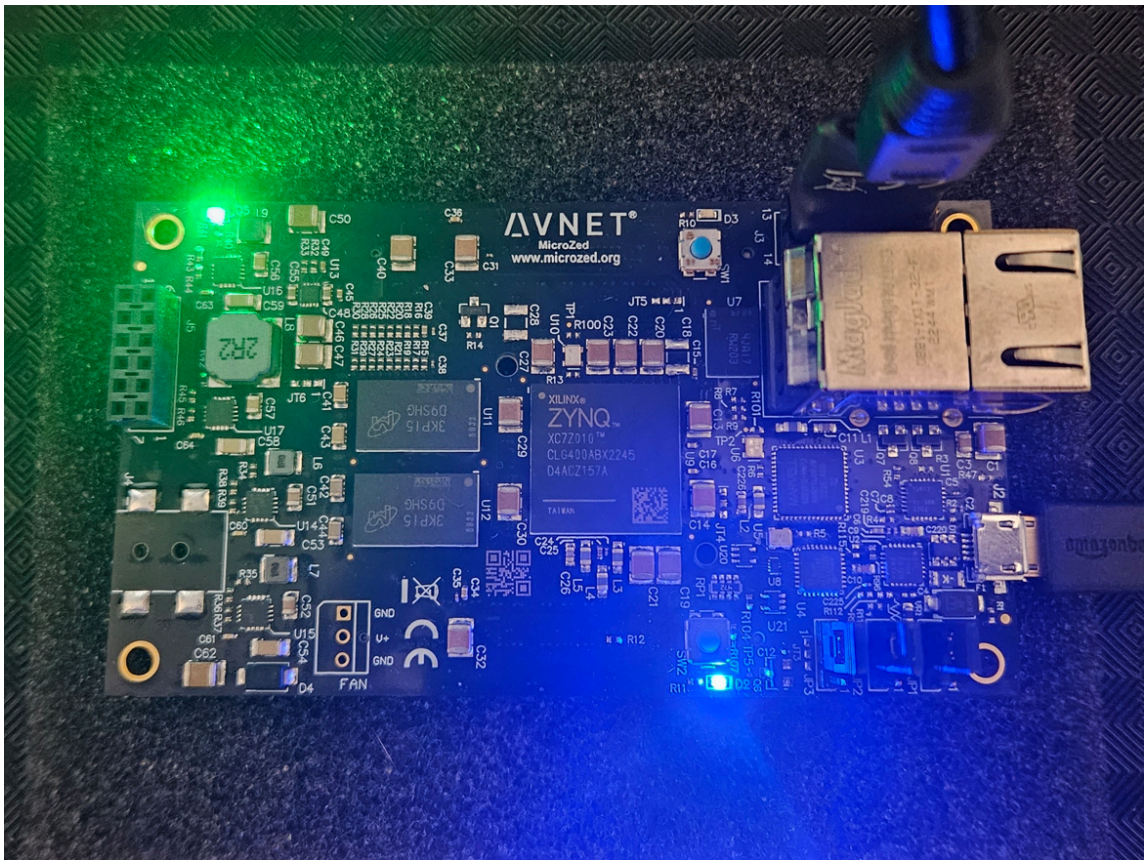


The test program simply prints a header for the software and then interfaces through embedded C to the custom-written hardware block from specialized addressing functions inside Vitis which performs concurrent addition and multiplication on the SoC. All operations from this embedded C program are executed on the custom and personalized hardware previously described.

After building the environment, programming to SoC, and running the application. The system results in the output of the image below.

```
----------SPACE TIME CARD MSD PROJECT 22257----------

DESCRIPTION:
   Software utilized in timecard to receive TOD
   from GNSS, export CAN communications,
   and calculate overall drift per earth day.

TEAM:
   Drew Schacke, Eva Czukkermann, Ian Dolfi,
   Luke Schrom, Nsadhu Muyinda, Tanner Smith

------------------------------------------------------------------------

Testing addition from a "custom" IP Hardware Block
   (A = 8)  +  (B = 18)  =  (SUM = 26)
Testing multiplication from a "custom" IP Hardware Block
   (A = 8)  *  (B = 18)  =  (PRODUCT = 144)
```

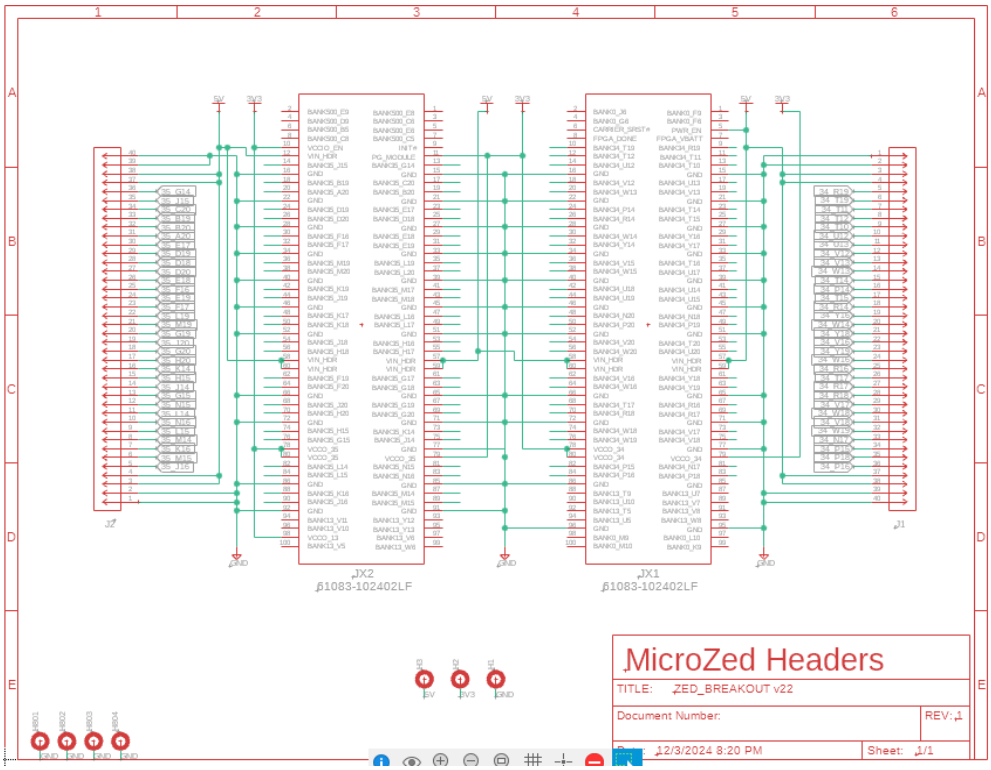
Once the physical SoC has been programmed, it is easy to tell since the blue light will illuminate to visualize the correct programming. The green light verifies that adequate power is being supplied.

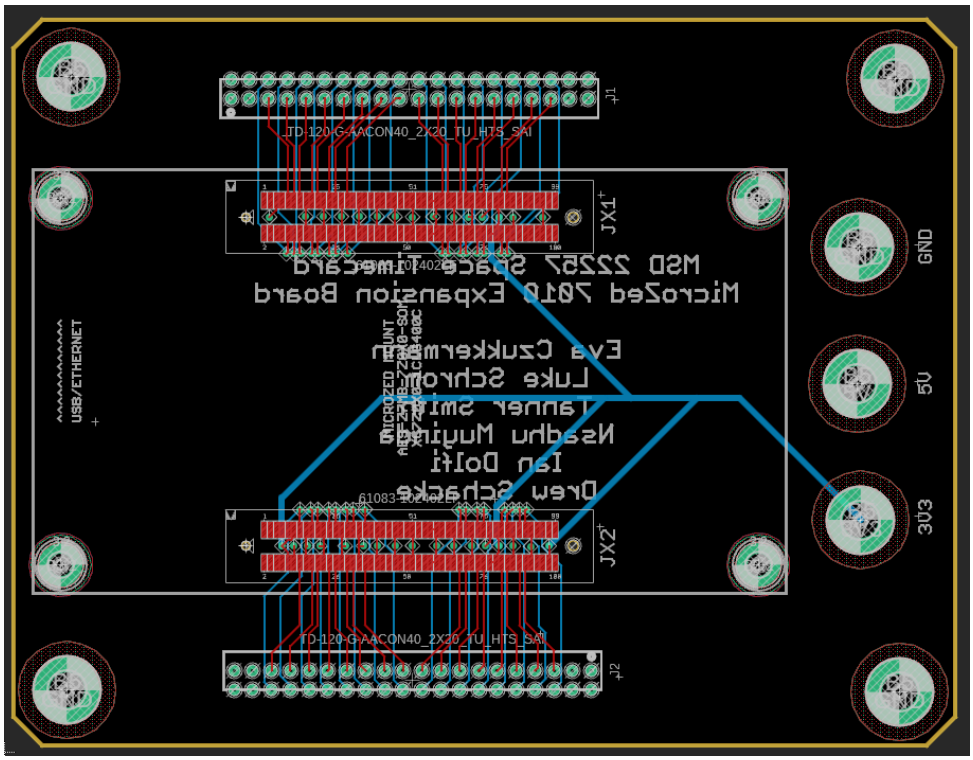


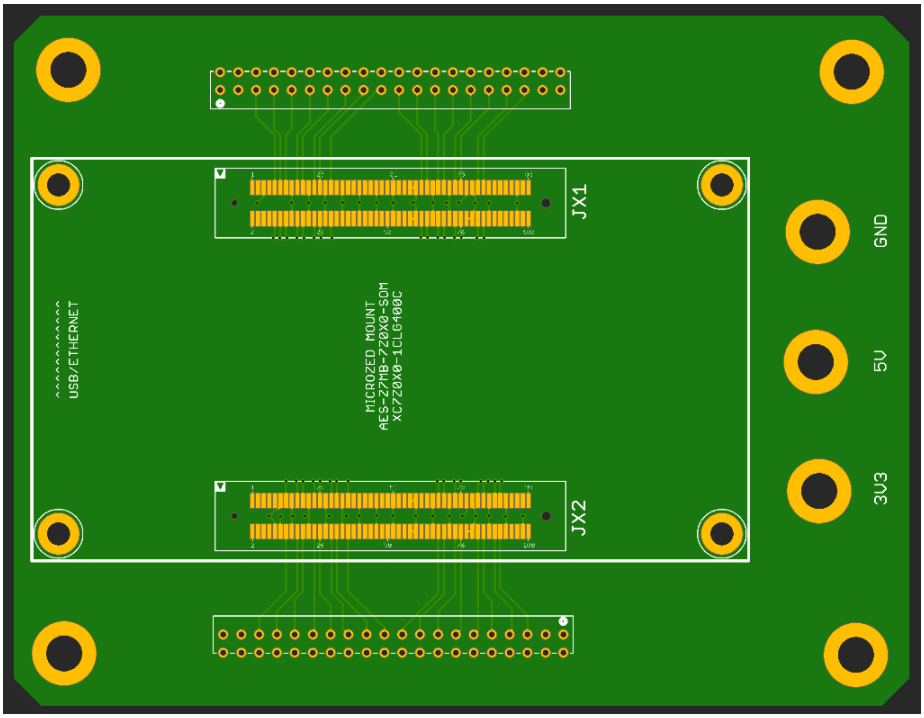
This environment is now completely set up and awaits implementation of the timecard digital design and software communication protocols and mathematics.

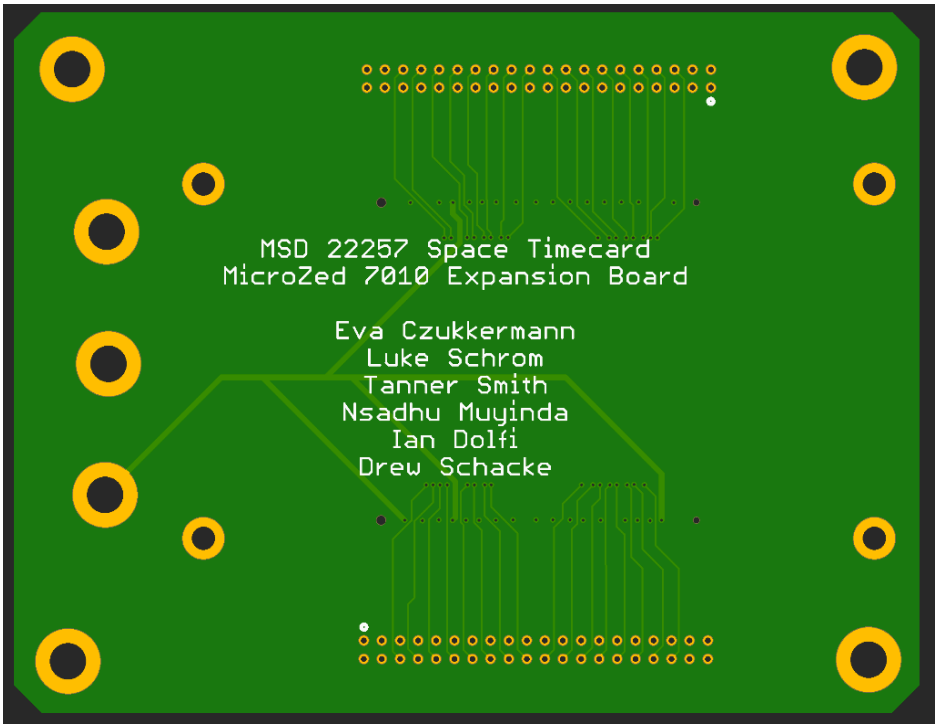
Drawings, Schematics, Diagrams:

MicroZed Expansion Board:



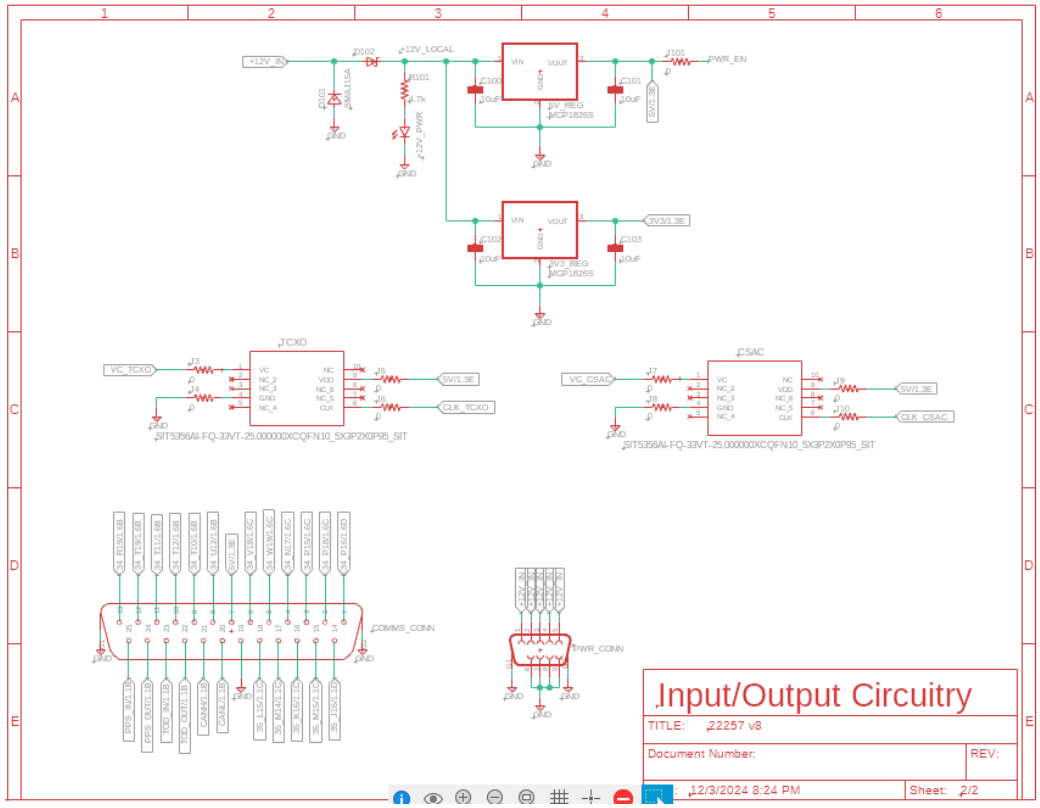


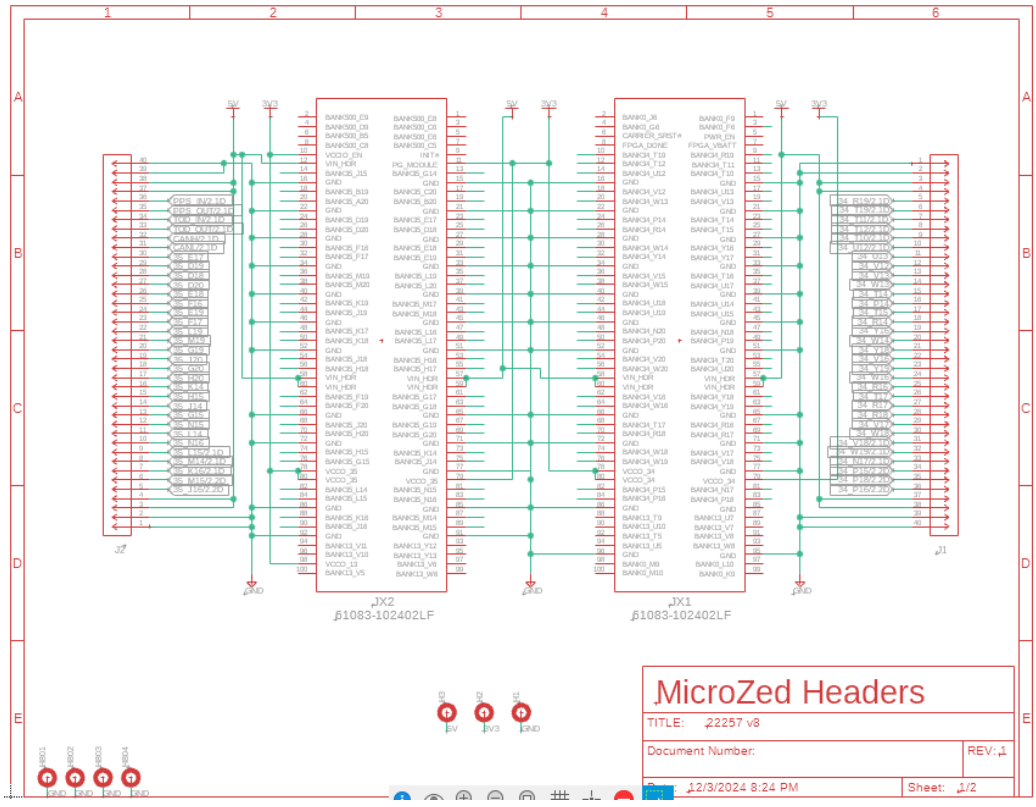


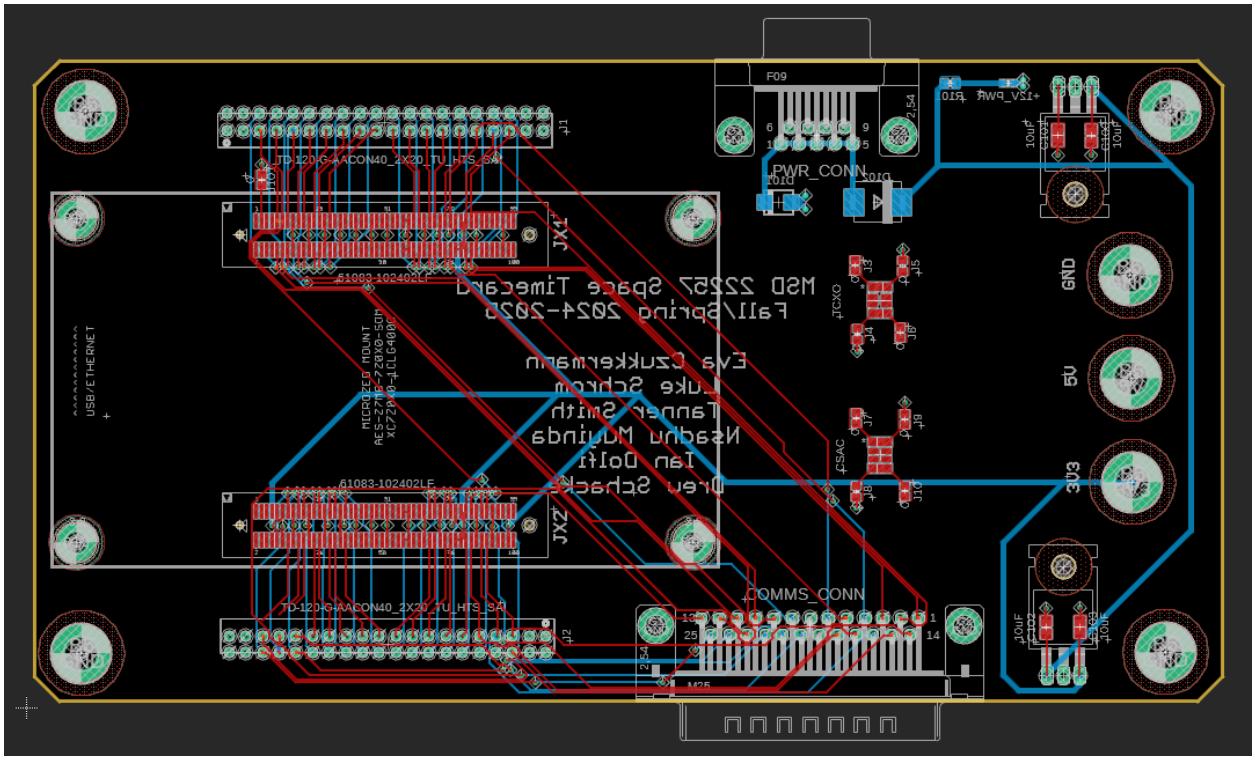


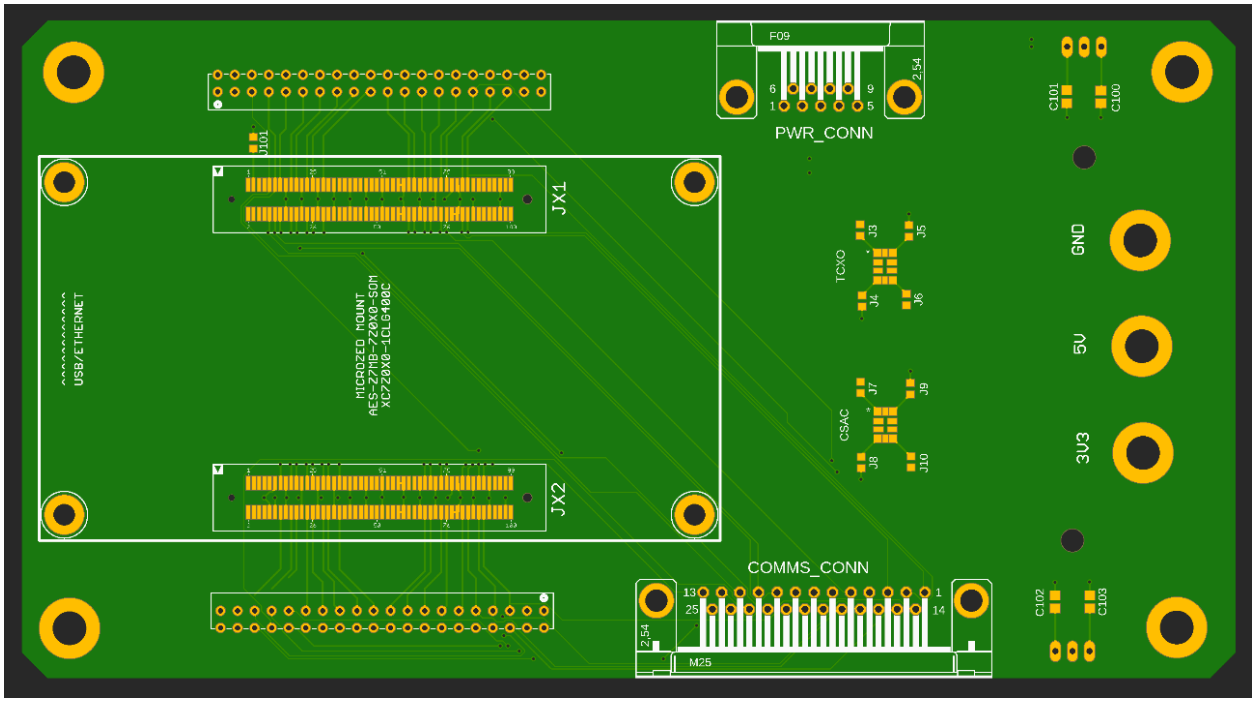
The MicroZed Expansion Board is a custom-built, 4-layer PCB utilized for breaking out the I/O pins of the MicroZed SoC utilized in our design. This board used the top/bottom for signals and components and inner layers for power and ground planes. This board will provide an efficient means of implementation and prototyping prior to producing the final Space Timecard.

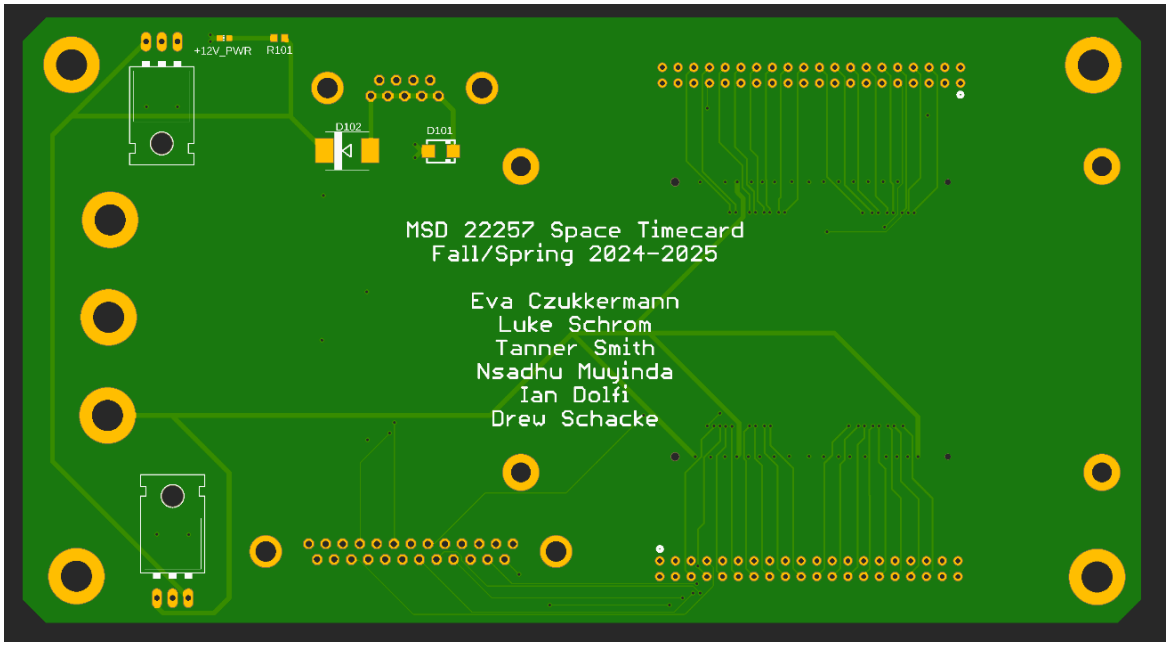
Space Time Card:





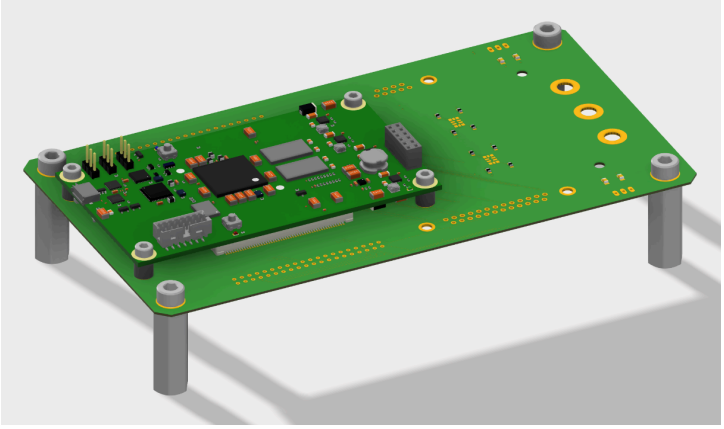

F09
PWR_CONN
JX1
JX2
GND
5V
3V3
COMMS_CONN
M25
TCXO
CSAC
USB/ETHERNET

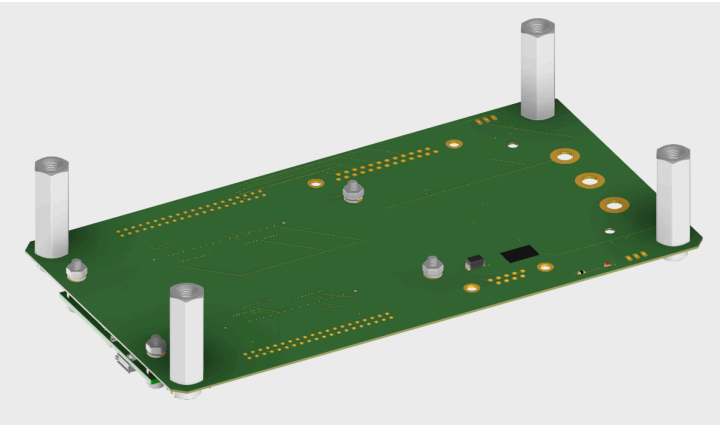

F09
PWR_CONN
JX1
JX2
GND
5V
3V3
COMMS_CONN
M25
TCXO
CSAC
USB/ETHERNET



The final 22257 Space Timecard is displayed above according to a 4-layer PCB where the top and bottom are used for signals and components and the inner layers are power and ground planes. This board offers an isolated +12V Rocket Power connector and a CubeSat motherboard communication connector. Ultimately, this board holds the MicroZed SoC, TCXO, and CSAC (interchangeably), voltage/power regulation, and optional I/O breakout to the SoC. The current implementation shows two TCXOs populated although one will be switched with the CSAC once the symbol/footprint is acquired. This design will serve as the final product considering some minor changes as prototyping continues.

Space Time Card 3D Rendering:





Above is a render of the board assembly. The SoC is situated on standoffs above the Time Card board and secured using screws, split lock washers, and nuts. The Time Card board is also situated on standoffs to allow for room for the components underneath. The SA-45 CSAC will be assembled directly on the Time Card board.

IPs from Meta: [Summary of IPs that could be implemented.]

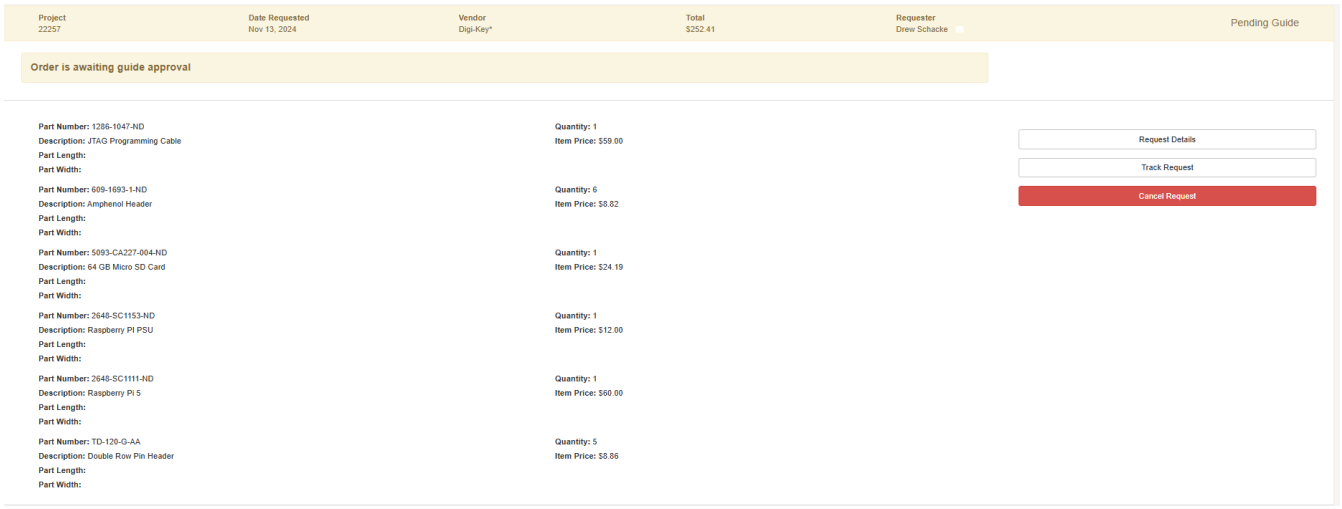
Bill of Material (BOM):

Team #:	P22257	Team Name:	Atomic Clock	Budget:	$4,000		Budget:	$4,000					KEY
Date:	10/17/2024	Document Owner:	Team Member	Total Projected:	$6,800.35		Total Spent:	$770.20					Ordering Early (10/31/24)
Revision #:	1	Date Last Updated: 12/03/2024		Remaining Projected:	-$2,800		Remaining:	$3,229.80					Ordering Early (11/13/24)

Subsystem	Team Part #	Item	Qty	Unit Cost	Total Cost	Total Spent with Shipping/Tarrifs/Tax	Vendor	Website	Ordered	Received	Notes
Electrical	00001	AES-Z7MB-7Z010-SOM-I-G/REV-H	2	$259.00	$518.00	$704.31	Avnet	https://www.avnet.com/shop/us/products/avnet-engineering-services/aes-z7mb-7z010-som-i-g-rev-h-3074457345644698176/#			Includes a $119.14 tariff
Electrical	00002	Ceramic Capacitor Kit 10pF ~ 10µF	1	$48.24	$48.24		DigiKey	https://www.digikey.com/en/products/detail/kemet/CER-ENG-KIT-29/5039067			
Electrical	00003	Resistor Kit 1k ~ 9.76k Ohm ±1% 1/10W	1	$38.30	$38.30		DigiKey	https://www.digikey.com/en/products/detail/stackpole-electronics-inc/KIT-RMCF0603FT-04/2773766			
Electrical	00004	Resistor Kit 10k ~ 97.6k Ohm ±1% 1/10W	1	$38.30	$38.30		DigiKey	https://www.digikey.com/en/products/detail/stackpole-electronics-inc/KIT-RMCF0603FT-05/2773767			
Electrical	00005	SiT5356AI-FQ233iT-10.000000F	3	$68.81	$206.44		SiTime	https://www.sitime.com/products/super-tcxo/sit5356#buy-now%20			
Electrical	00006	SiT5356AI-FQ133iT-10.000000F	3	$68.81	$206.44		SiTime	https://www.sitime.com/products/super-tcxo/sit5356#buy-now%20			
Electrical	00007	SiT5356AI-FQ033iT-10.000000F	3	$68.81	$206.44		SiTime	https://www.sitime.com/products/super-tcxo/sit5356#buy-now%20			
Electrical	00008	CSAC SA-45 090-03240-003	1	$5,377.76	$5,377.76		MicroChip	https://www.microchip.com/en-us/product/csac-sa45s#document-table			Client purchased outside of these funds!
Electrical	00009	MCP251863T-H/SS	10	$2.34	$23.40		DigiKey	https://www.digikey.com/en/products/detail/microchip-technology/MCP251863T-H-SS/15985649			
Electrical	00010	SMT Breakout PCB for SOIC-28 or TSSOP-28 - 3 Pack!	1	$4.95	$4.95		Adafruit	https://www.adafruit.com/product/1208			
Electrical	00011	Large Premium Solderless Breadboard	1	$19.95	$19.95		Adafruit	https://www.adafruit.com/product/443			
Electrical	00012	Hook-up Wire Spool Set - 22AWG Solid Core - 6 x 25 ft	1	$15.95	$15.95		Adafruit	https://www.adafruit.com/product/1311			
Test Setup	00013	2648-SC1153-ND Raspberry Pi 5	1	$60.00	$60.00		DigiKey	https://www.digikey.com/en/products/detail/raspberry-pi/SC1111/21658261			
Test Setup	00014	2648-SC1111-ND Raspberry Pi 5 PSU	1	$12.00	$12.00		DigiKey	SC1153 Raspberry Pi	Power Supplies - External/Internal (Off-Board)	DigiKey			
Test Setup	00015	5093-CA227-004-ND 64GB Micro-SD card	1	$24.19	$24.19		DigiKey	https://www.digikey.com/en/products/detail/kahasra-electronic-solutions-limited/CA227-004/21089320			
Test Setup	00016	609-1693-1-ND Amphenol Headers	6	$8.82	$52.92		DigiKey	https://www.digikey.com/en/products/detail/amphenol-cs-fci/61083-102402LF/10020077?mpckid=2b5da0f773a26a542cf1aa04720a6b04			
Electrical	00017	1286-1047-ND JTAG Programming Cable	1	$59.00	$59.00		DigiKey	https://www.digikey.com/en/products/detail/digilent-inc/410-299/5015666?s=N4IgTCBcDaIFiBUCCBxAtACQMoGYAEACzE4D2ASkQJYC21AIgxHZI4DCIARgDYCmIAugF8gA			
Electrical	00018	TD-120-G-AA Double Row Pin Header	5	$8.86	$44.30		DigiKey	https://www.digikey.com/en/products/detail/samtec-inc/TD-120-G-AA/10218852			
Test Setup	00019	474-GPS-14986 GNSS Antenna	1	$12.95	$12.95	$65.89	Mouser	https://www.mouser.com/ProductDetail/SparkFun/GPS-14986?qs=bveeYqUIh0OBAxR%252BCRKDBA%3D%3D			
Test Setup	00020	474-GPS-18037 GNSS Reiever	1	$44.95	$44.95		Mouser	https://www.mouser.com/ProductDetail/SparkFun/GPS-18037?qs=sGF71Wb1eUMf7i0ct1O1A%3D%3D			
Electrical	00021	SENSOR ANALOG -40C-150C SC70-5 Temp Sensor	5	$0.26	$1.32		DigiKey	https://www.digikey.com/en/products/detail/texas-instruments/TMP235A4DCKR/8597481			

The most up-to-date BOM is shown above and the master document is linked at the end of this paper. Major highlights from this BOM show 2 MSD completed and received orders, one CSAC order provided by the client, and one order in the process of purchase. Considering items purchased and acquired, we have spent a total of $770.20 leaving $3,229.80 available from the $4000 provided. The client purchase came to a total of $5,792.33 resulting in $6,562.53 invested.

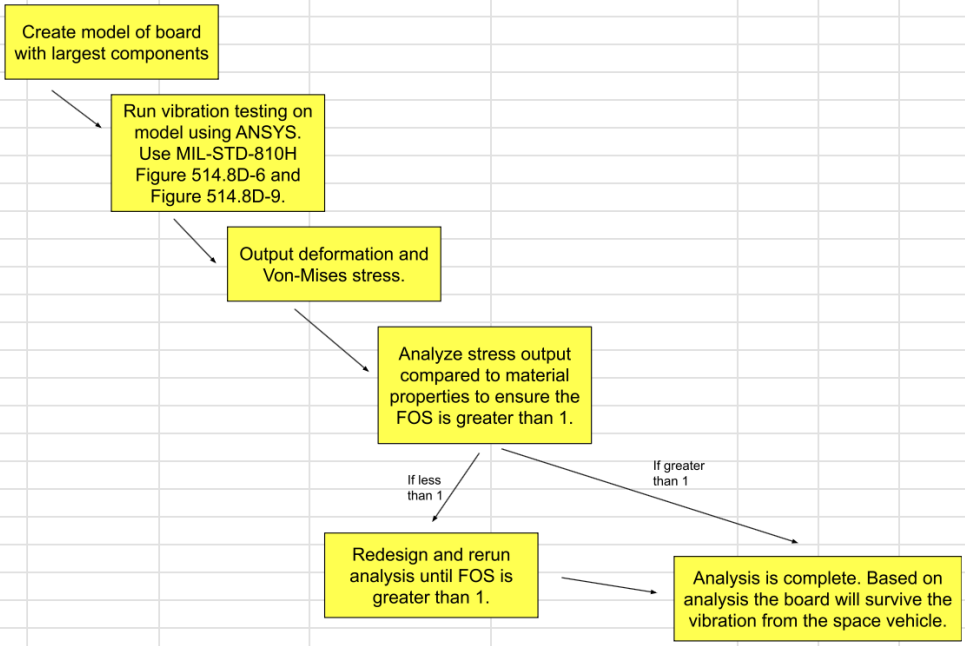
Any/all long lead items have been purchased and acquired leaving minimal risk due to logistics and purchasing. The pending order is attached directly below. Note that the order was submitted on Nov 13, 2024 yet has not been ordered yet due to extreme technical issues plaguing the MSD PICS website.


Project
22257
Date Requested
Nov 13, 2024
Vendor
Digi-Key*
Total
$252.41
Requester
Drew Schacke
Pending Guide
Order is awaiting guide approval
Part Number: 1286-1047-ND
Description: JTAG Programming Cable
Part Length:
Part Width:
Quantity: 1
Item Price: $59.00
Part Number: 609-1693-1-ND
Description: Amphenol Header
Part Length:
Part Width:
Quantity: 6
Item Price: $8.82
Part Number: 5093-CA227-004-ND
Description: 64 GB Micro SD Card
Part Length:
Part Width:
Quantity: 1
Item Price: $24.19
Part Number: 2648-SC1153-ND
Description: Raspberry PI PSU
Part Length:
Part Width:
Quantity: 1
Item Price: $12.00
Part Number: 2648-SC1111-ND
Description: Raspberry PI 5
Part Length:
Part Width:
Quantity: 1
Item Price: $60.00
Part Number: TD-120-G-AA
Description: Double Row Pin Header
Part Length:
Part Width:
Quantity: 5
Item Price: $8.86
Request Details
Track Request
Cancel Request

Test Plans:

Vibe Analysis Test using Ansys (ER 4):

Subsystem/ Function/ Feature Name:	Vibe Analysis							
Date of Test:								
Performed By:	Eva Czukkermann							
Tested By:	Eva Czukkermann							
Concluded Condition of meeting Engineering Specification:								
I. TESTING SPECIFICATION								
Specification Number	Importance	Source	Function	Specification (Metric)	Unit of Measure	Marginal Value	Ideal Value	Comments/Status
S4	3		System	IAW MIL-STD-810H Figure 514.8D-6 and Figure 514.8D-9	N/A	Yes	Yes	
II. EQUIPMENT REQURIED								
Specification Number	Equipment or Instrumentation required							
S4	Ansys, CAD							
III. DATA COLLECTION STRATEGY								
Specification Number	Data acquisition strategy							
S4	The goal of this test is to ensure the board can survive the frequencies/vibrations it will experience while assembled in a space vehicle.							
III. TESTING FLOWCHART								



MIL-STD-810H Figure 514.8D-6 and Figure 514.8D-9 reference Table 514.8D-IV for levels (W_1, W_2, f_0, f_1). See the figures and table below:

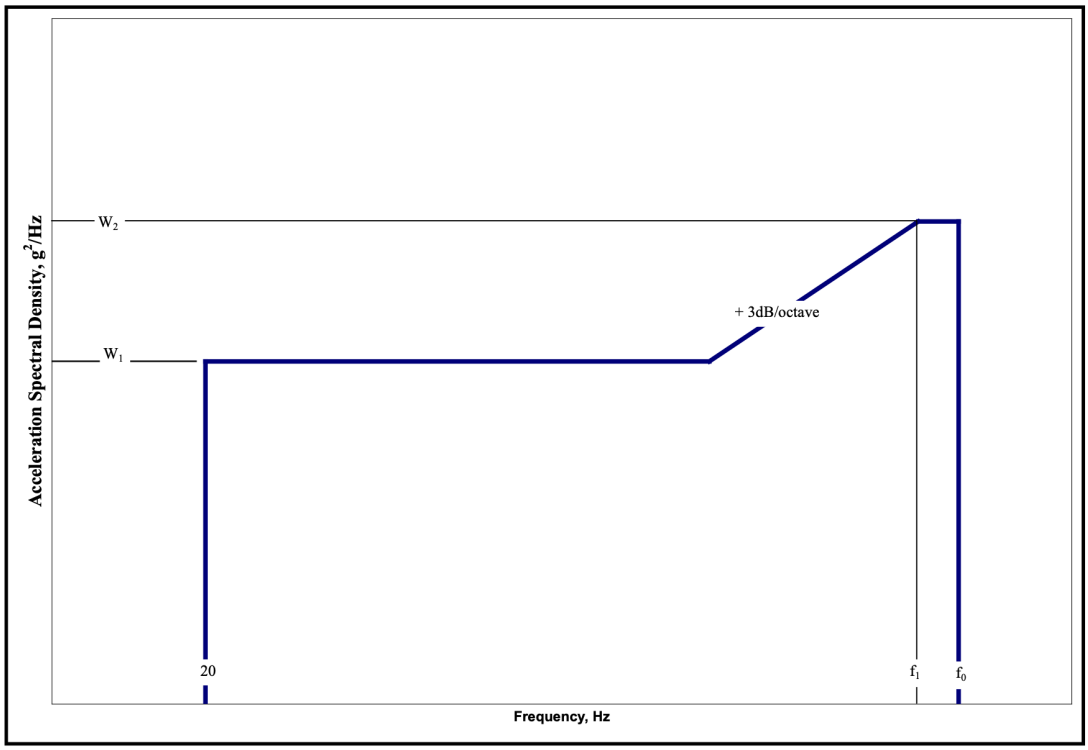


Figure 514.8D-6. Category 15 - Jet aircraft store vibration response.

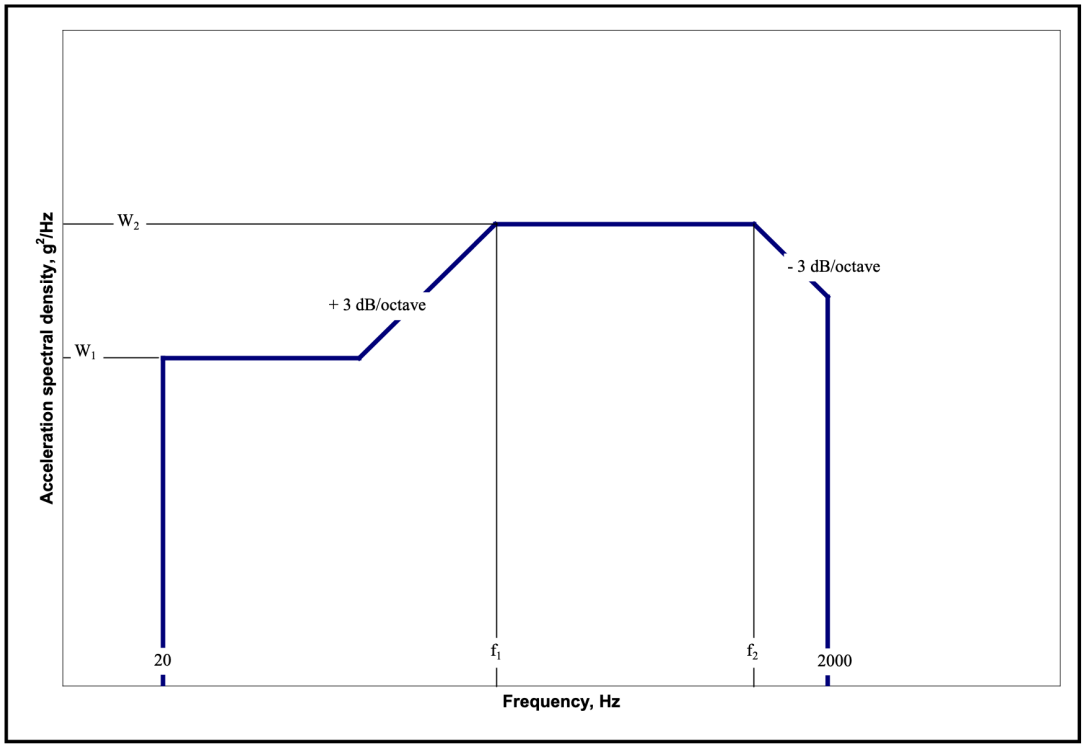


Figure 514.8D-9. Category 16 - Jet aircraft store equipment vibration exposure.

Table 514.8D-IV. Category 15 - Jet aircraft external store vibration exposure.

$W_1 = 5 \times 10^{-3} \times K \times A_1 \times B_1 \times C_1 \times D_1 \times E_1$; (g^2/Hz) 1/

$W_2 = H \times (q/\rho)^2 \times K \times A_2 \times B_2 \times C_2 \times D_2 \times E_2$; (g^2/Hz) 1/

$M \leq 0.90$, $K = 1.0$; $0.90 \leq M \leq 1.0$, $K = -4.8 \times M + 5.32$; $M \geq 1.0$, $K = 0.52$ 2/

$f_1 = 10^5 C(t/R^2)$, (Hz) 3/, 4/, 5/; $f_2 = f_1 + 1000$, (Hz) 3/:

$f_0 = f_1 + 100$, (Hz) 6/, 7/

Configuration	Factors		Configuration	Factors	
Aerodynamically clean	A_1	A_2		B_1	B_2
Single store	1	1	Powered missile, aft half	1	4
Side by side stores	1	2	Other stores, aft half	1	2
Behind other store(s)	2	4	All stores, forward half	1	1
Aerodynamically dirty 8/	C_1	C_2		D_1	D_2
Single and side by side	2	4	Field assembled sheet metal		
Behind other store(s)	1	2	fin / tail cone unit	8	16
Other stores	1	1	Powered missile	1	1
	E_1	E_2	Other stores	4	4
Jelly filled firebombs	1/2	1/4			
Other stores	1	1			

M – Mach number.
H – Constant = 5.59 (metric units) ($= 5 \times 10^{-5}$ English units).
C – Constant $= 2.54 \times 10^2$ (metric units) (= 1.0 English units).
q – Flight dynamic pressure (see Table 514.8D-V) – kN/m^2 (lb/ft^2).
ρ – Store weight density (weight/volume) - kg/m^3 (lb/ft^3).
Limit values of ρ to $641 \leq \rho \leq 2403$ kg/m^3 ($40 \leq \rho \leq 150$ lb/ft^3).
t – Average thickness of structural (load carrying) skin - m (in).
R – Store characteristic (structural) radius m (in) (Average over store length).
= Store radius for circular cross sections.
= Half or major and minor diameters for elliptical cross section.
= Half or longest inscribed chord for irregular cross sections.

1/ – When store parameters fall outside limits given, consult references.
2/ – Mach number correction (see Annex B).
3/ – Limit f_1 to $100 \leq f_1 \leq 2000$ Hz
4/ – Free fall stores with tail fins, $f_1 = 125$ Hz
5/ – Limit length ratio to: $0.0010 \leq C(t/R^2) \leq 0.020$
6/ – $f_o = 500$ Hz for cross sections not circular or elliptical
7/ – If $f_0 \geq 1200$ Hz, then use $f_0 = 2000$ Hz

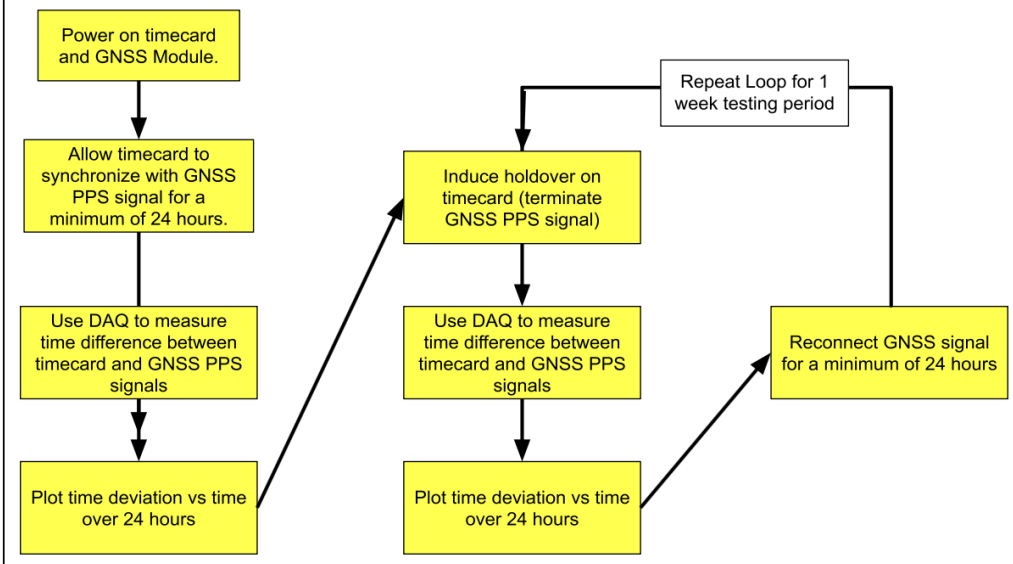
8/ – Configurations with separated aerodynamic flow within the first ¼ of the store length. Blunt noses, optical flats, sharp corners, and open cavities are some potential sources of separation. Any nose other than smooth, rounded, and gently tapered is suspect. Aerodynamics engineers should make this judgment.

Representative parameter values

Store type	Max q		ρ		f_1	f_2
	kN/m^2	(lb/ft^2)	kg/m^3	(lb/ft^3)	Hz	Hz
Missile, air to ground	76.61	(1600)	1602	(100)	500	1500
Missile, air to air	76.61	(1600)	1602	(100)	500	1500
Instrument pod	86.19	(1800)	801	(50)	500	1500
Dispenser (reusable)	57.46	(1200)	801	(50)	200	1200
Demolition bomb	57.46	(1200)	1922	(120)	125	1100
Fire bomb	57.46	(1200)	641	(40)	100	1100

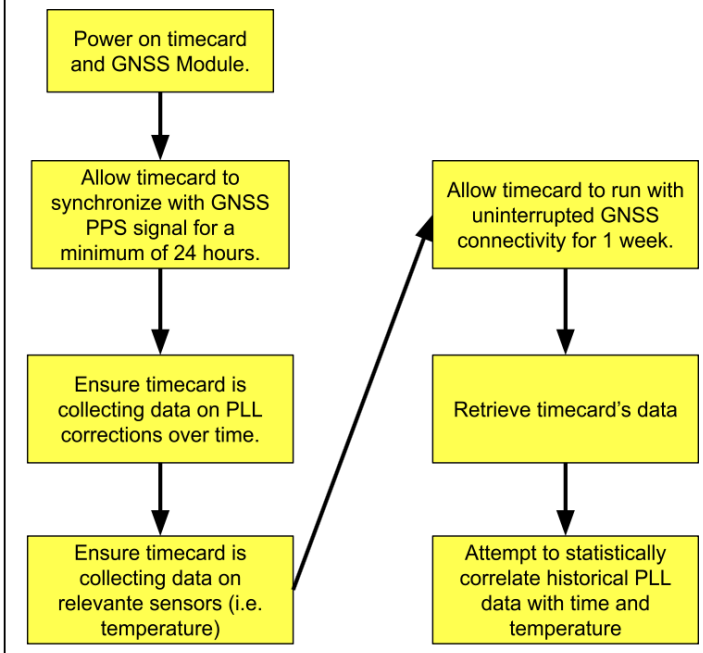
See this document for an explanation of why these figures and the table are used. They will be used to perform a structural analysis to ensure the time card can withstand the vibrations it will experience.

Holdover Accuracy Verification (ER 9):

Subsystem/ Function/ Feature Name:	Holdover Accuracy							
Date of Test:								
Performed By:								
Tested By:								
Concluded Condition of meeting Engineering Specification:								
I. TESTING SPECIFICATION								
Specification Number	Importance	Source	Function	Specification (Metric)	Unit of Measure	Marginal Value	Ideal Value	Comments/Status
S9	1	ITU-T G.8273	Timing	Timekeeping drift *shall* not exceed 30 nanoseconds.	Nanoseconds per Earth day	1	30	
II. EQUIPMENT REQURIED								
Specification Number	Equipment or Instrumentation required							
S9	Oscilloscope, GNSS Module							
III. DATA COLLECTION STRATEGY								
Specification Number	Data acquisition strategy							
S9								

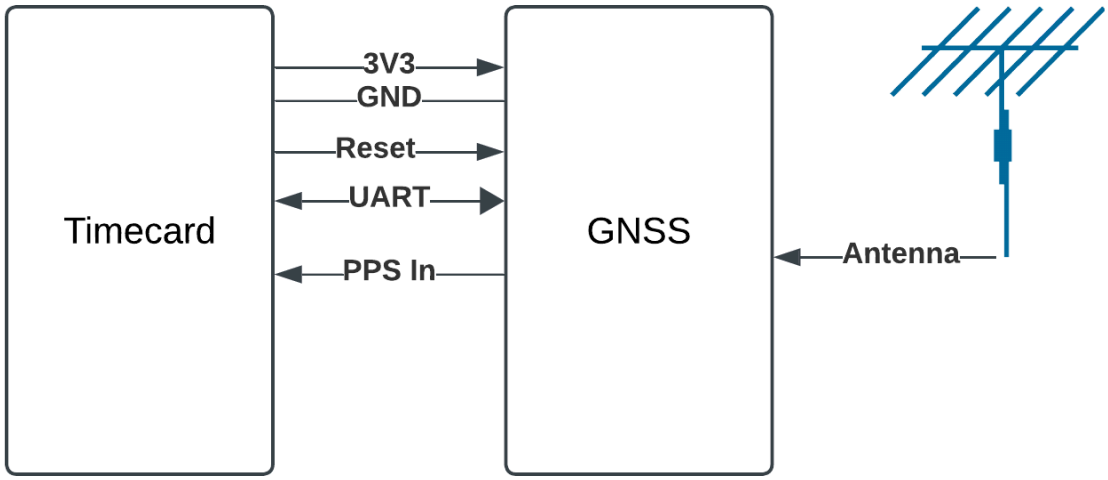
This test allows us to verify the metrics of ER9: “Timekeeping drift shall not exceed 30ns per day”. This test is performed by monitoring the time error of the timecard while doing the following: sync the timecard to GPS for a day, cut off GPS communications for a day, resync for a day, and repeat as needed (we chose one week total for this test). The metric obtained from this test is the maximum time error of the timecard during any given time of this experiment except for the initial synchronization.

Functionality of Historical Data (precursor to ER 9):

Subsystem/ Function/ Feature Name:			Functionality of Historical Data					
Date of Test:								
Performed By:								
Tested By:								
Concluded Condition of meeting Engineering Specification:								
I. TESTING SPECIFICATION								
Specification Number	Importance	Source	Function	Specification (Metric)	Unit of Measure	Marginal Value	Ideal Value	Comments/Status
Precursor to S9	3		Timing Corrections		N/A	Yes	Yes	
II. EQUIPMENT REQURIED								
Specification Number	Equipment or Instrumentation required							
Precursor to S9	Computer, GNSS Module, temperature DAQ							
III. DATA COLLECTION STRATEGY								
Specification Number	Data acquisition strategy							
S9								

While designing our system, we must be able to statistically correlate the frequency errors of the clock to both time and temperature. This test serves as a way to obtain the data to perform this statistical correlation. The timecard runs for a long time (1 week is chosen) while already locked to GPS and constantly in communication with GPS, and data is collected on the frequency adjustments that are needed for this clock over time, along with temperature data.

GNSS and Clock Drift Test:



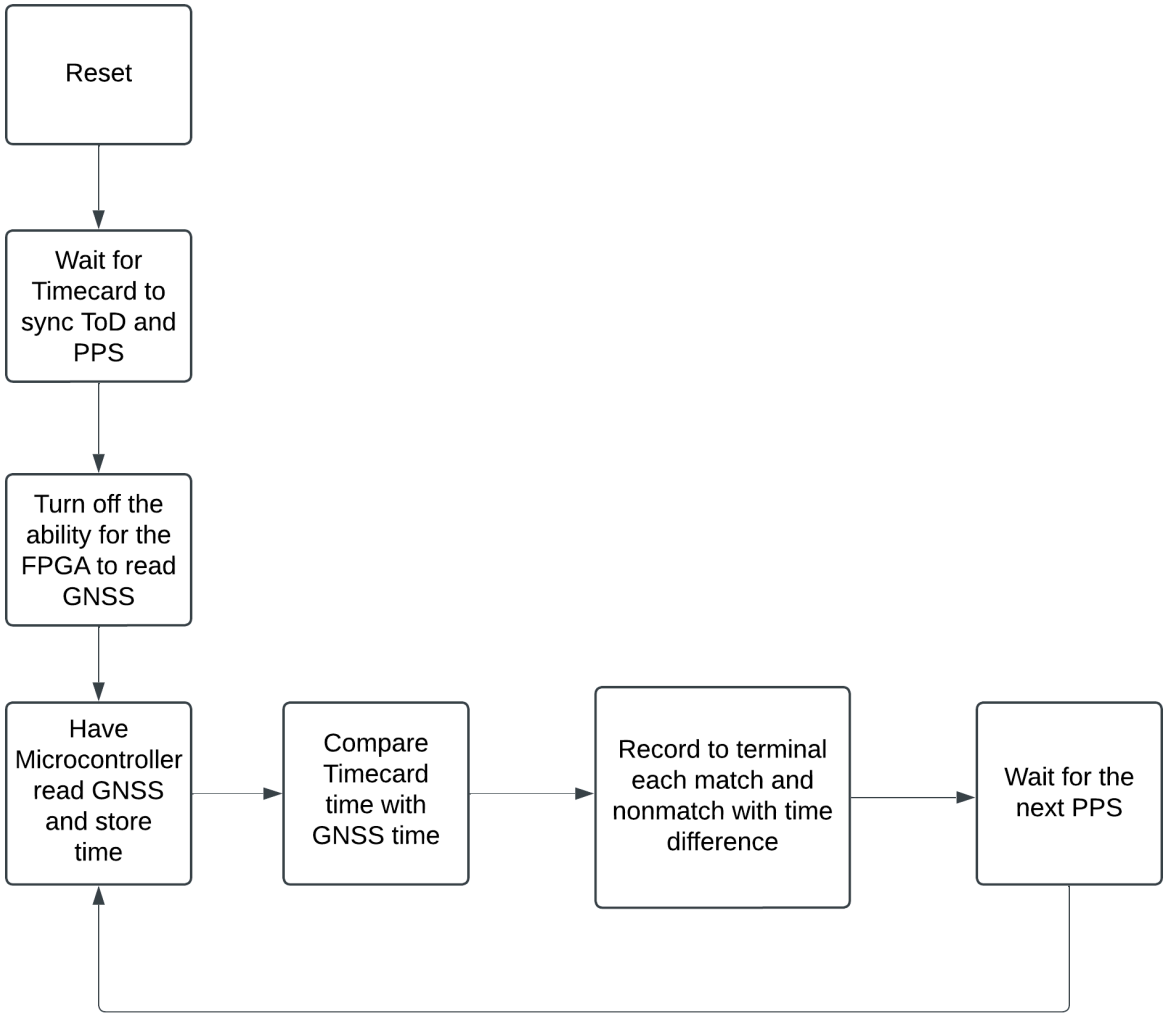
The timecard is designed to interface with a GNSS module using the ToD (Time of Day) Slave IP block from the Open Compute Time Card Project. By integrating a GNSS receiver and antenna, the system can receive signals from Galileo.

A program can be implemented on the microcontroller to read and compare the ToD values from the GNSS module and the timecard. These comparisons can be output to the terminal and logged into an error file for later analysis. This functionality allows for real-time monitoring of synchronization accuracy between the GNSS and the timecard.

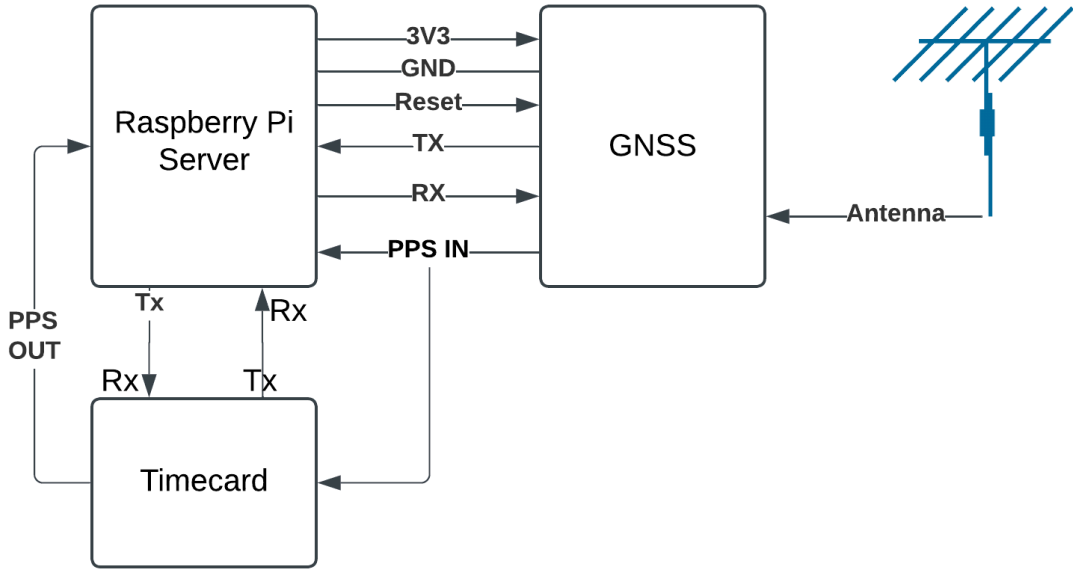
System Details:

- **Power Supply:** The GNSS module is powered directly via the SoC GPIO.
- **Data Communication:** Data from the GNSS is transmitted to the timecard over UART.
- **Signal Configuration:** The ToD Slave block is configured to use the Galileo signal provided by the GNSS.

The program is designed to operate autonomously over extended periods, provided the system remains undisturbed. This capability facilitates testing of the system's long-term performance, particularly under conditions without an active GNSS connection. Such testing will provide valuable insights into system behavior and stability over time.


Reset
Wait for Timecard to sync ToD and PPS
Turn off the ability for the FPGA to read GNSS
Have Microcontroller read GNSS and store time
Compare Timecard time with GNSS time
Record to terminal each match and nonmatch with time difference
Wait for the next PPS

GNSS Test Plan



Goal of test:

The purpose of this test is to configure a Raspberry Pi to collect data from a GNSS module and relay it to the timecard. In its initial implementation, the data transfer will occur over UART, with provisions for transitioning to CAN once the interface becomes available.

System Overview:

- **Power Supply:** The GNSS module is powered by a 3.3V supply from the Raspberry Pi.
- **Communication:** The GNSS communicates bidirectionally with the Raspberry Pi via UART.
- **PPS Signal:** The GNSS provides a PPS (Pulse Per Second) signal, which serves as the PPS IN signal for the timecard. This is utilized by the DPLL on the timecard for synchronization.
- **Sync Time**: Have a server that can pull the time from Galileo.
- **Programming Language:** The server on the Pi will be written in Python because the GitHub repos for the GPIO are out of date.

Operation Details:

1. **Frequency Selection:**
   The Raspberry Pi sends a command to the GNSS module specifying the desired frequency for operation. For instance, the frequency 1575.42 MHz will be configured to receive data from the Galileo system.
2. **Data Collection:**
   - The GNSS transmits data to the Raspberry Pi in NMEA format over UART.

    - Relevant NMEA messages include:
        - **GPRMC:** Recommended Minimum Navigation Data.
        - **GPZDA:** Time and Date.
    - The Raspberry Pi parses these messages to extract essential data.
3. **Data Processing and Relay:**
   The extracted data is either relayed as-is or processed into a preferred format before being sent to the timecard. The timecard uses this information to establish the reference time.

Goals:

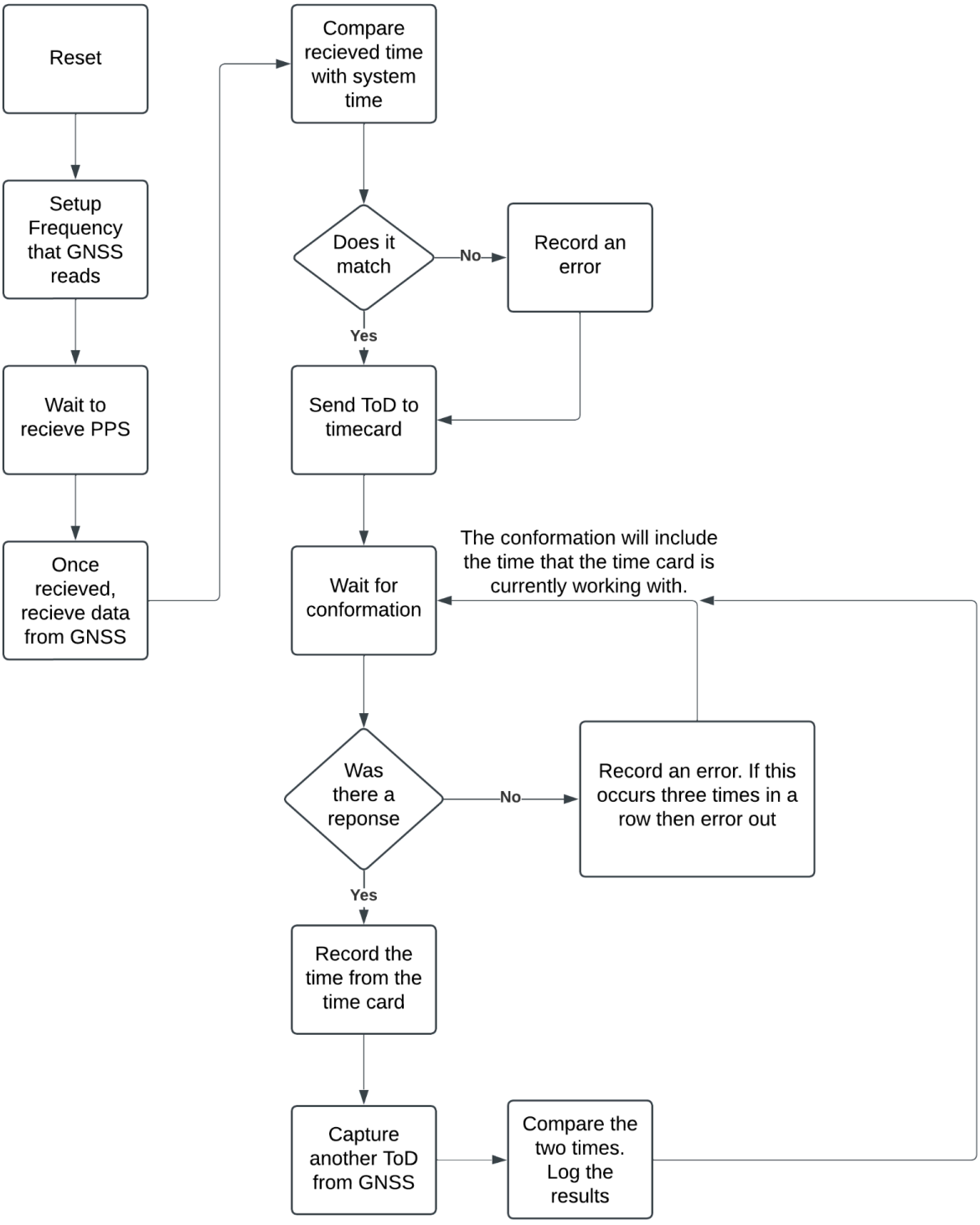
- Successfully gather and relay GNSS data to the timecard.
- Configure the system for a seamless transition to CAN communication.
- Ensure accurate synchronization with the PPS signal.

From looking up examples online of people doing similar things I have found this GitHub repo: https://github.com/jacobjhansen/RTCM-Pi
I can use this as an example for extracting data from the NMEA messages and relaying them to the timecard. Work smarter, not harder.
The reason why we are using a Raspberry Pi as our server is because we could just set up the interface for GNSS with the timecard but IT isn't allowing us all to remote into the computer that we have our Vivado environment set up on. The Raspberry Pi offers a mobile way of developing a test setup.

Required Materials:

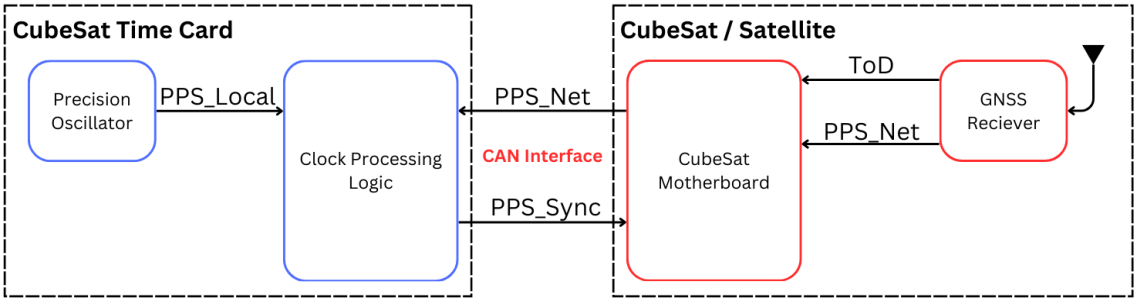
- Raspberry Pi
- Sparkfun GNSS Receiver Breakout Board
- GNSS Antenna

Outline of Server's Operation:



System Design and Flowcharts/System Block Diagram:

High-Level System diagram:



The figure above represents the high-level diagram showing the interaction between the CubeSat Satellite and the time card. This representation is very similar to the standard off-the-shelf timecard, with the major difference identified as the GNSS receiver is onboard the satellite, not the timecard.

Board Functionality Diagram:

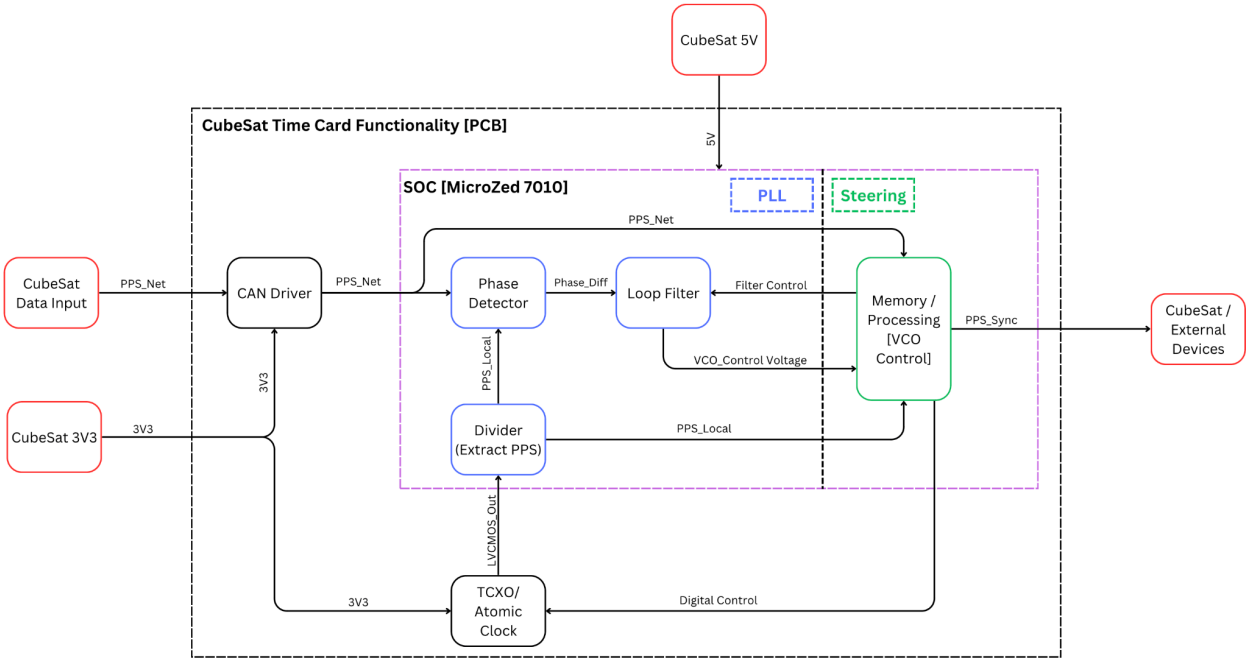


Diagram Link:
https://www.canva.com/design/DAGU9_wbepU/3GgiWLj0Og3cwYhZnoNWGQ/view?utm_content=DAGU9_wbepU&utm_campaign=designshare&utm_medium=link&utm_source=editor

Functional Block Explanations:

Phase Detector:
Takes two PPS inputs, and produces output that is proportional to the phase difference between the two input signals. Traditionally output a pulse width proportional to the phase difference.

Loop Filter:
Reduces noise, improves stability, shapes the transient response, and controls the phase margin. Outputs a VCO control signal, used to tune/adjust the characteristics of the local TCXO/Atomic clock.

Steering Process:
Produce coefficients to alter/tune the TCXO or Atomic clock to match phase with the PPS produced with the GNSS receiver.

Divider:
Receives the LVCMOS signal produced from the TCXO/Atomic Clock. Reduces the frequency of the input by a specific ratio. This must be an accurate and precise process.

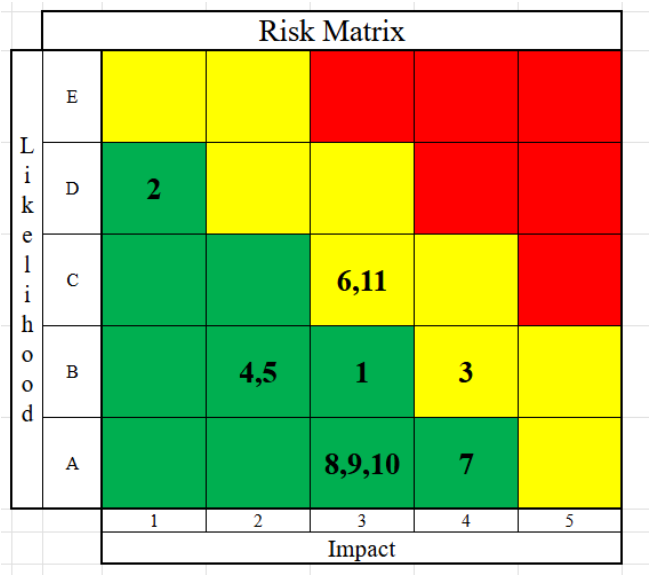
Phase Lock Loop [PLL]:
Often used for frequency synthesis, clock recovery, and motor controllers. The PLL receives two input signals, identified as the reference and feedback signals. In this case, the feedback signal is represented by the locally produced PPS. Similarly, the reference signal is represented by the PPS produced from the GNSS. The difference in phase between the two input signals is extracted through the use of a phase detector. This difference is then fed into a loop filter that produces a correlated value to the phase difference. This, in turn, gives a variable that can be used to “tune” the local TCXO/Atomic clock, to be in phase with the external network GNSS-produced PPS.

Risk Assessment:

Risk Assessment remains on track with no additional items or changes since the last phase review. The current risk of long lead items is not anticipated due to early actions taken to acquire yet remains on the assessment due to the possibility of future purchases or unforeseen circumstances. The full spreadsheet is attached at the end of this paper.

Team #:	22257	Team Name:	Space Time Clock
Date:	12/3/24 17:47	Document Owner:	
Revision #:	1	Project Budget	$ 10,000.00

ID	IF Statement	THEN Statement	Cause	Likelihood	Cost Impact	Schedule Impact	Importance	Action to Minimize Risk
1	IF timecard is exposed to higher EMI than expected	THEN received data will become corrupted/incorrect	Strong electromagnetic fields and RF from other satillites	B	2	3	B3	Provide error flag and power cycle device
2	IF communication protocol experiences bit flip and interference	THEN stored data will become corrupted/incorrect	Cosmic Rays	D	1	1	D1	Force developmet to parity and error detection schemes in firmaware for all communications
3	IF open source programs do not work as described	THEN design is set behind due to designing additional programs/functions/blocks	Unpredicted Factors	B	3	4	B4	Our team will produce our own open source programs for any of those which do not work as described
4	IF vaccum causes solderpoints to flake and cause shorts	THEN a high current flow (short) can occur, causing component damage	Material Science Factors	B	2	2	B2	Utilize Sn62Pb37 solder rather than lead-free (tin) solder
5	IF vaccum causes PCB to outgas	THEN surrounding components and systems experience unexpected debris which may lead to opens/shorts	Material Science Factors	B	2	2	B2	Place PCBs through vaccum cycles prior to using
6	IF cubesat vibrates more than design specifications	THEN connection would be lost between the timecard and motherboard	Rough Launch, Space Debris Collision	B	2	3	B3	Develop option to mechanically strengthen the electrical connectors
7	IF the timecard loses power	THEN the oscillator and all other components will not function	CubeSat loses power	A	3	4	A4	No resolution, other than power cycle once CubeSat regains power
8	IF Displacement Damage Dose occurs	THEN non-ionizing energy loss occurs causing device and/or component degradation	Energy deposition by impinging radiation	A	3	3	A3	If CubeSat is not protected, develop protection barrier
9	IF Single Event Effects (Single Event Latch Up) occur	THEN communication packets will contain incorrect data	Singular, stray, energetic particles	A	3	3	A3	If CubeSat is not protected, develop protection barrier. Can additional implement strongery parity detection and schemes
10	IF Atomic Oxidation occurs	THEN electrical connection including solder points may lose conductivity	Presence of strong oxidizing agents	A	3	3	A3	If CubeSat is not protected, develop protection barrier
11	IF parts have lead times over a month	THEN parts may not be in hand when required to prototype	Supply/Demand	C	2	3	C3	Order significantly early than expected, if not possible, look for other vendors or options.



Open Items:

CURRENT PHASE OPEN ITEMS

Open Design Items:

- Currently no open design items, ahead of schedule, and working through implementation.

FUTURE PHASE OPEN ITEMS

Open Implementation Items:

- Implementation of feasibility design from MATLAB/SIMULINK into hardware via the SoC's Programmable Logic (PL) and Processing System (PS)
- Final implementation of PCB after prototype verification
- Create an emulated CubeSat motherboard on our Raspberry Pi 5

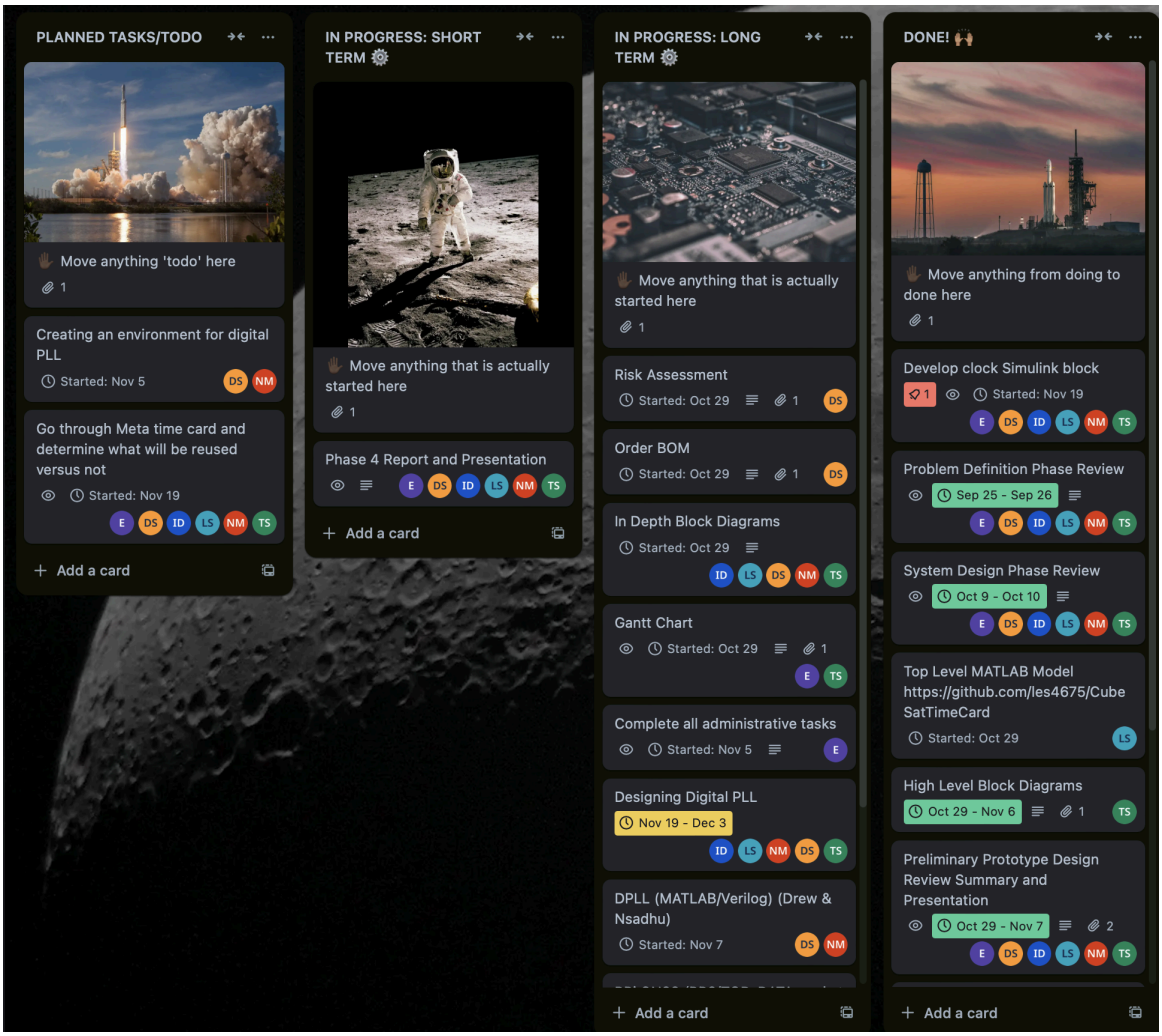
Open Testing Items:

- Functionally verify implemented design
- Demonstrate integration within our emulated CubeSat using the Raspberry Pi 5

Project Management:

- What went well?
    - We continued to make progress with the materials that we have. We ordered our most expensive component - the CSAC SC45.
- What could be improved?
    - Not all team members have defined tasks to work on during each meeting. We also don't stick to the tasks we listed in our Trello board tasks.
- What do we commit to improving in the next Sprint?
    - We commit to staying more organized with taskings and ensure we know our short-term and long-term goals.

Trello Board Snippet:



[Link to Trello Board]

Revised August 2024

Design Review Materials:

Pre-read: Summary Report and Presentation

- Agenda
    - Discuss progress made in the Detailed Design Phase
- Presenters
    - Eva Czukkermann, Ian Dolfi, Nsadhu Muyinda, Drew Schacke, Luke Schrom, Tanner Smith

Notes from Review with Action Items: Design Review Notes

Supporting Documentation:

Presentation
System Block Diagram
Trello Board
Risk Assessment
Gantt Chart
Requirements and Testing
Bill of Materials
Vibe Figure Explanation

Goals for next phase:

Upon exiting MSD 1 and entering MSD 2, the “Build and Test Prep” phase will be entered. As mentioned previously we are technically ahead of schedule, and we will be striving to our momentum. Our goals for the next phase include:

- Furnish and order the breakout board for the MicroZed 7010 SoM.
- Furnish the time card complete PCB.
- Assemble and test functionality of the SoM, CSAC, and TCXO.
- Furnish existing test plans and develop new tests as required.

Gantt Chart:

Link: Gantt Chart

Personal Goals / Assignments:

Drew Schacke:

Creating an environment for digital PLL within MicroZed 7010 [Vivado]. Designing Digital PLL. Circuit and complete PCB design.

Eva Czukkermann:
Prepare documents needed throughout Phase 5.
Complete all administrative tasks.
Assist with interfacing between MicroZed 7010 [SoC] and multiple TCXO/Atomic Clock breakout boards.

Ian Dolfi:
Interfacing between MicroZed 7010 [SoC] and multiple TCXO/Atomic Clock breakout boards. Designing Digital PLL. Continue benchmarking/Analysis on MATLAB for PLL design. Circuit and PCB design.

Luke Schrom:
Assist with the design of digital PLL. Benchmarking/analysis on MATLAB for PLL design. Development of GNSS Raspberry Pi testbench. Circuit and PCB design.

Nsadhu Muyinda:
Assist in creating an environment for digital PLL within MicroZed 7010 [Vivado]. Designing Digital PLL.

Tanner Smith:
Interfacing between MicroZed 7010 [SoC] and multiple TCXO / Atomic Clock breakout boards. Designing Digital PLL. Identify parts needed for testing and benchmarking time card. Circuit and PCB design.