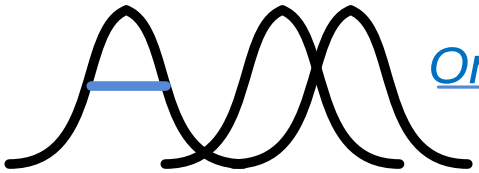


# *Photogrammetry for Non-Invasive Terrestrial Position/Velocity Measurement of High-Flying Aircraft*

## *Part 6A & B: Completing the Hardware*

James A Crawford

### *Synopsis*

*A lot has happened since my last project-update. The Part B material picks up in §12. More specifically:*

- *Precision optical encoder assemblies have been designed, fabricated, and installed on the telescope azimuth and elevation axes.*
- *The elevation axis approach has been changed to now incorporate an 80:1 harmonic drive for improved torque without backlash.*
- *A Raspberry Pi4 HQ imager (12.3 MP, Sony IMX477 sensor) and associated optical train has been designed, fabricated, and installed on the telescope, and the first astronomical images captured.*
- *A finder telescope has been designed and fabricated to make manual sight-alignment easier.*
- *Telescope rings and other associated hardware have been designed and fabricated to mount the finder telescope on the main telescope.*
- *An adjustable (interim) tripod has been designed, fabricated, and assembled to host the complete telescope for near-term testing.*
- *The TI DSP configuration has been modified to accommodate 3-phase motor drivers for the two axes.*
- The DSP code of Part V was re-written to move real-time optical encoder sample read operations from the DSP's CPU to the DSP's CLA[1] thereby ensuring uninterrupted synchronous operation and more efficient use of the DSP's computing resources.
- *And in Part B, C# GUI and TI software have been modified to accommodate two 3-phase DRV8301 motor drivers and two optical encoders. This required a second DRV8301* Booster Pack to be hosted on the TMS320F28379D DSP to simultaneously support azimuth and elevation axes operations
  - SPI pin-outs of Part V subsequently had to be rearranged.
  - The TMS320F28379D only has 3 SPI ports and each DRV8301 motor Booster Pack unfortunately requires a dedicated SPI port. This only left one SPI port available to read the two optical encoders involved.
- A hardware/software multiplexing scheme was developed to use the remaining SPI port in a time-multiplexed manner to circumvent the issue.

[1] CLA stands for control law accelerator.

 

- *A Raspberry Pi4 was configured with the TMS320F28379D signal processor to (i) support real-time telescope image capturing, (ii) ultimately perform future image processing, and (iii) host a GPS receiver so that precision time keeping can be easily provided to the TI DSP. The resultant multi-processor configuration is shown in Figure 1.*
- *A new PCB was designed and fabricated for routing signals between the TI DSP and all of the external peripherals (optical encoders, 3-phase motors, RP4, power supply, etc.) and housed in a new chassis with front panel connectorized interfaces.*
- *The Part B material picks up in §12.*

*For those who may be unfamiliar with my on-going project, a synopsis of earlier project installments follows:*

1. *Part I provided a simple introduction to the big-picture objectives for this multi-phase project.*
2. *Part II first looked at telescope mounts, ultimately focusing on the azimuth-elevation type mount for the project.* The basic mathematics for dealing with 3-phase DC motors (e.g., Clarke and Park transformations) were introduced, along with the first ingredients for modeling and controlling the DC motors in a precision manner. The TMS320F28379D *Launchpad* hardware platform from Texas Instruments was selected to host the motor control algorithms.
3. *In Part III, most of the attention was focused on the mechanical design, fabrication, and assembly of the telescope mount. The detailed design of the hardware changed appreciably from the first concept as better approaches were recognized during the detailed design effort. A first-look at low rotational speed cogging torque was also conducted using the actual (azimuth) 3-phase DC motor. An early version of the optical encoder mounting method was also presented (which was subsequently changed).*
4. *In Part IV, attention was directed to (i) the mechanical drive details for the elevation axis finally settling on a 25:1 belt-drive step-down approach and (ii) electrical interfacing to the optical encoders. Substantial details for electrical interfacing with the AEAT-9000 optical encoders for the azimuth and elevation axes were presented. An Arduino Mega2560 was used as an interim step for the interface before moving this task over to the much more capable TMS320F28379D digital signal processor. Later on in Part I, however, the elevation axis approach was changed to a harmonic drive approach for much better torque and general performance.*
5. *Part V saw completion of a new and improved optical encoder mounting approach along with culmination of the supporting Arduino Mega2560 interface electronics. The associated functionality and more was subsequently hosted on the TMS320F28379D DSP with much better performance achieved. These steps also necessitated the development of a C# PC-based GUI.* An alignment method was perfected, enabling stand-alone encoder alignment without the telescope mount involved, permitting subsequent installation on the telescope mount without requiring any subsequent encoder re-alignment.

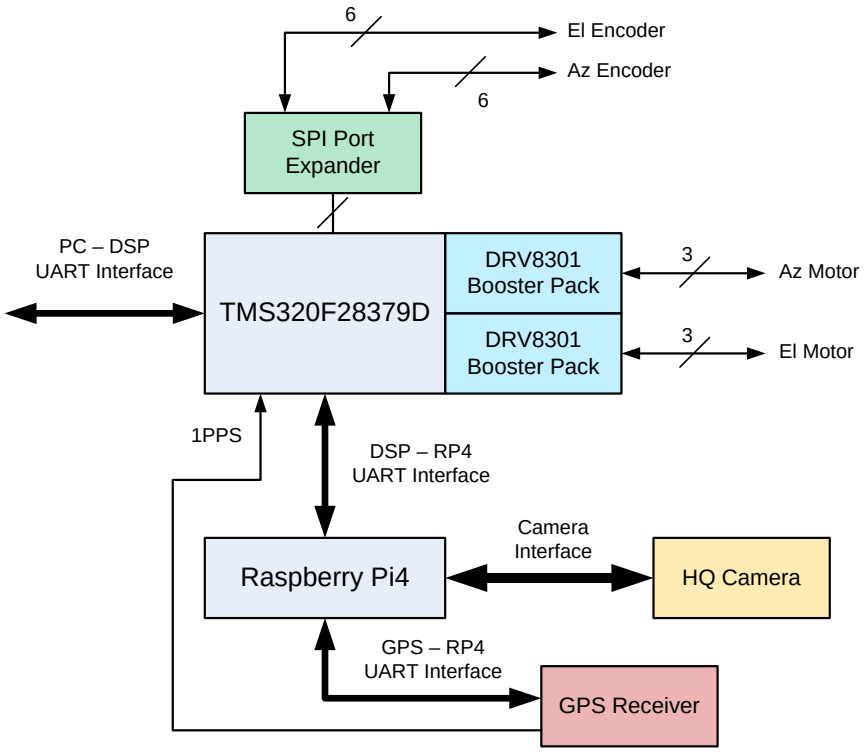


**Figure 1** Multi-processor approach going forward to accommodate (i) Az & El 3-phase motor control, (ii) Az & El optical encoders, and (iii) Raspberry Pi4 to host an HQ image sensor and GPS receiver[2]

# 1 Optical Encoder Completion

The optical encoders reported on in Part V [1] were aligned in a stand-alone manner and then installed on the azimuth and elevation axes as shown in Figure 5 respectively Figure 6. The flex-couplers combined with the rigidity of the precision x-y adjustment stages maintained the stand-alone alignment after each encoder was installed into the telescope. I fabricated additional encoder mounts for the future as shown in Figure 2.

# 2 Elevation Axis Drive

As mentioned in the opening comments, I completely redesigned my approach for the elevation axis drive. Although I had invested quite a bit of effort into the pulley-based system as described in Part IV [4], I opted to go with the more mechanically robust harmonic drive option.

I decided to purchase the main harmonic drive components on Ebay (Figure 3) primarily because my tooling was not appropriate for fabricating the flex-spline portion and the time spent on this portion of the project had already become excessive. When I purchased the harmonic drive components, the price was reasonable (roughly $150) but since that time, prices have gone through the roof and this particular model size is now selling for $900 and more. If I use harmonic drives in future project versions, this topic will have to be revisited because of cost.

As shown in Figure 8 through Figure 11, I designed and fabricated a housing to hold the harmonic drive components along with additional bearings and clamps. In general, the harmonic drive components must be suitably captured to permit free axial rotation while not permitting any longitudinal movement of the components along the axle.

The assembly and final harmonic drive configuration are shown in Figure 12 through Figure 15.

[2] U27771_Part6_Figures.vsd.

 

**Figure 2** Replicated optical encoder mounts to support two additional telescope efforts planned for the future

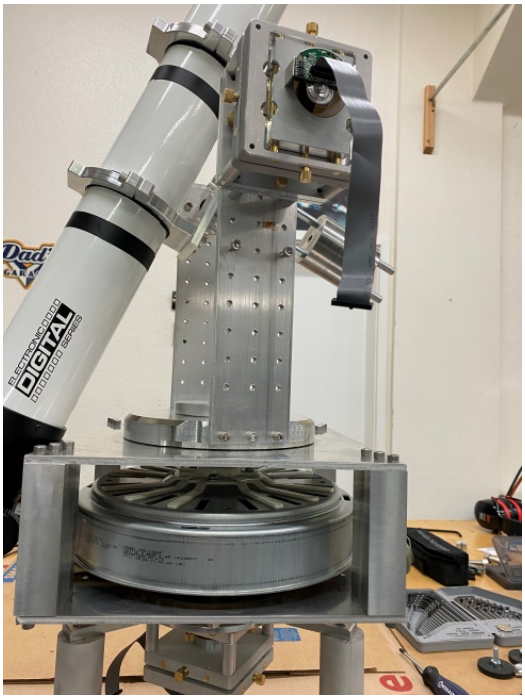

**Figure 3** Optical encoders just mounted to the telescope mount

**Figure 5** Azimuth encoder installed below the azimuth axis motor

**Figure 4** In the workshop

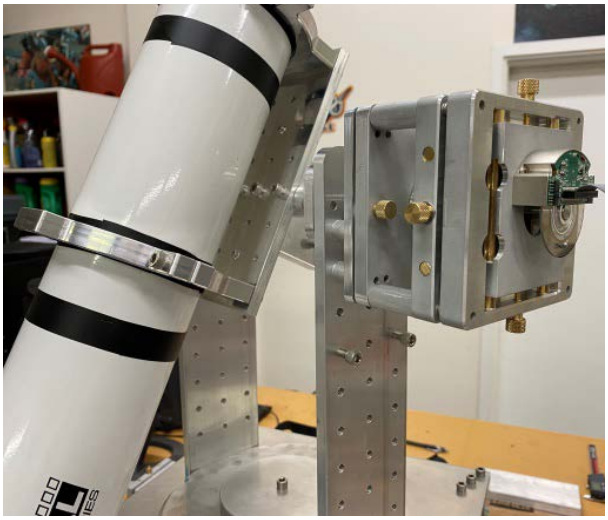

**Figure 6** Elevation encoder installed on one side of the axis

**Figure 7** Primary harmonic drive components purchased on Ebay, further augmented with an axle clamp.

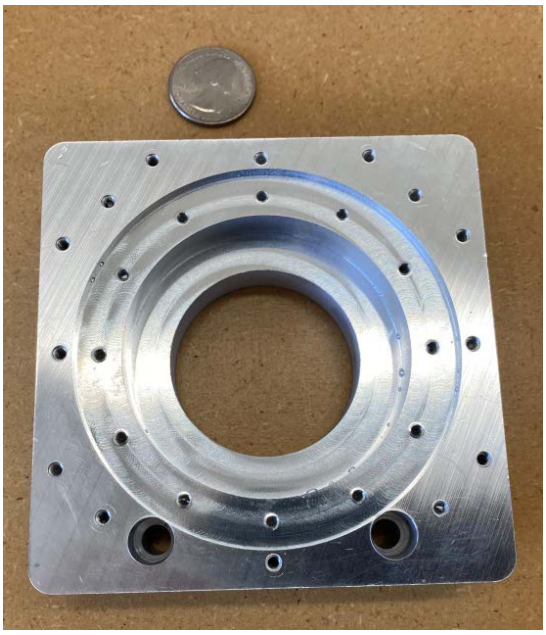

**Figure 8** Part of the fabricated harmonic drive housing used to secure the components into their proper place

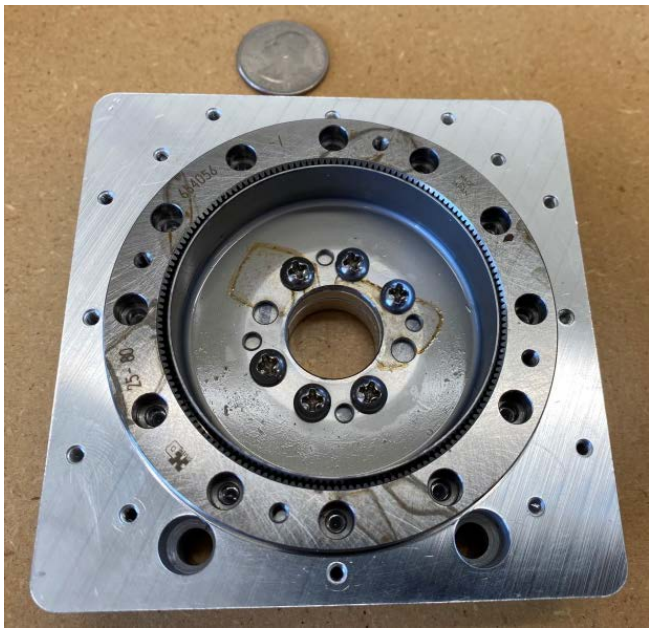

**Figure 9** Two of the three major harmonic drive components temporarily installed into the housing of Figure 8

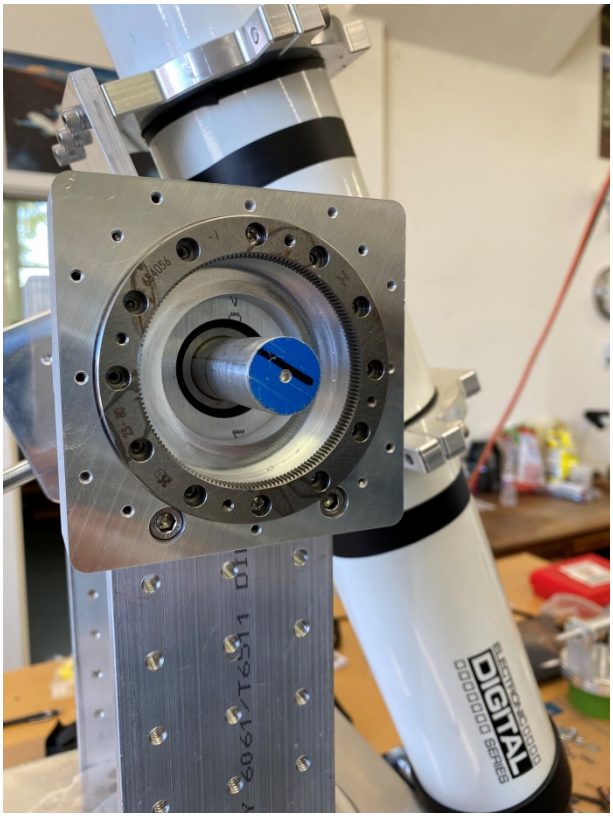

**Figure 10** Harmonic drive assembly into the telescope mount

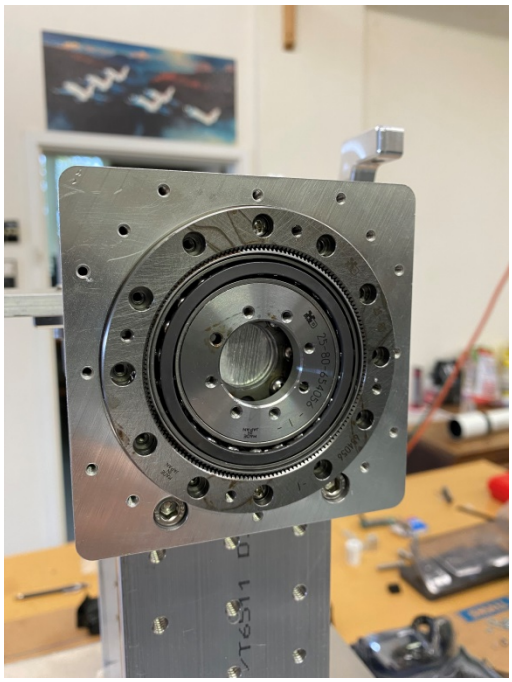

**Figure 11** Interim assembly step

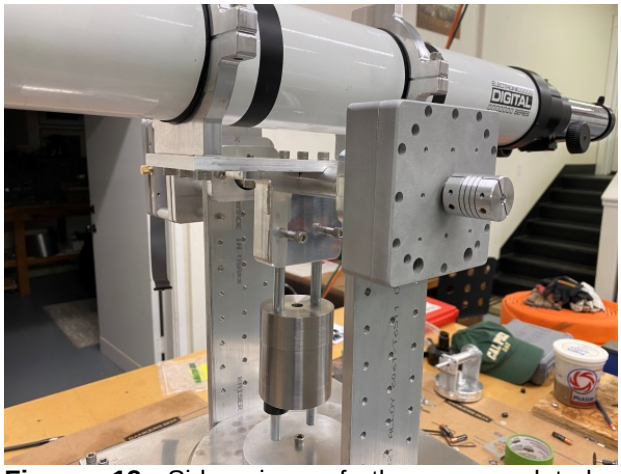

**Figure 12** Side view of the encapsulated harmonic drive before installing the motor

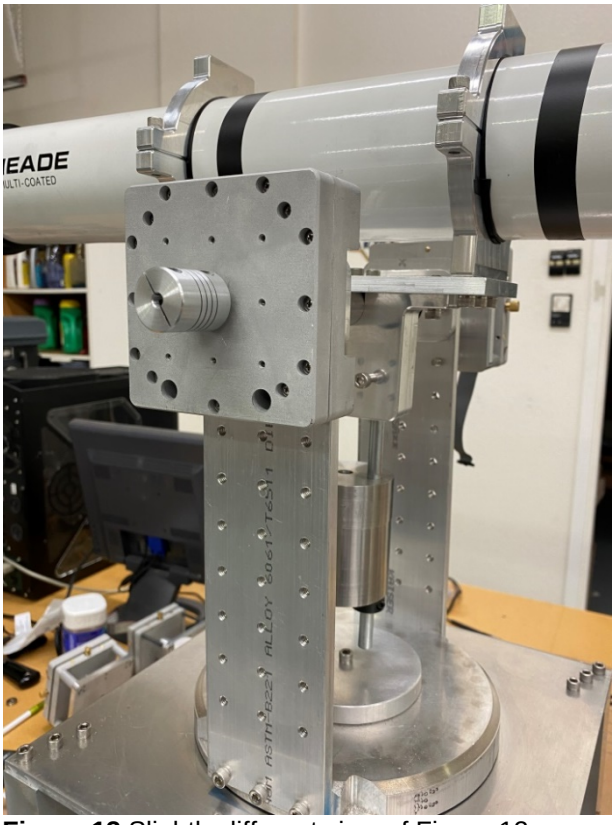

**Figure 13** Slightly different view of Figure 12

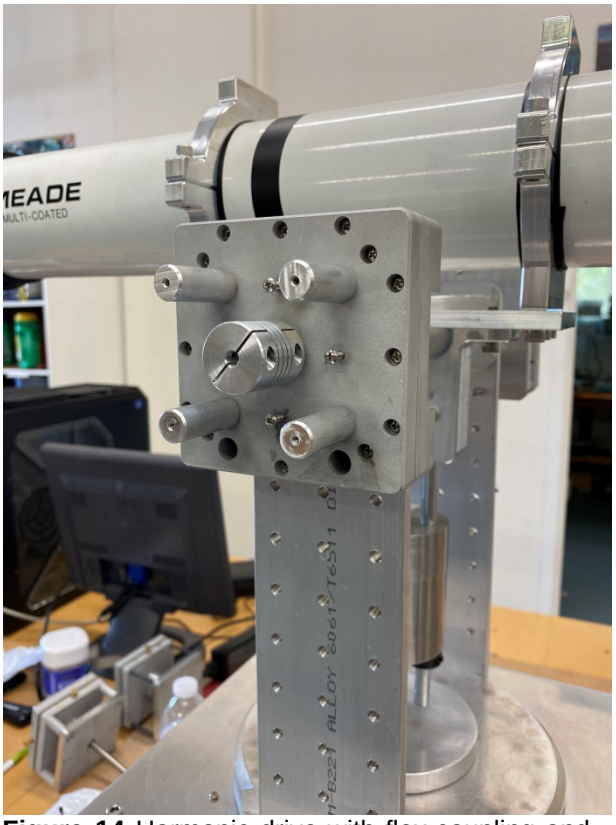

**Figure 14** Harmonic drive with flex-coupling and motor offset posts

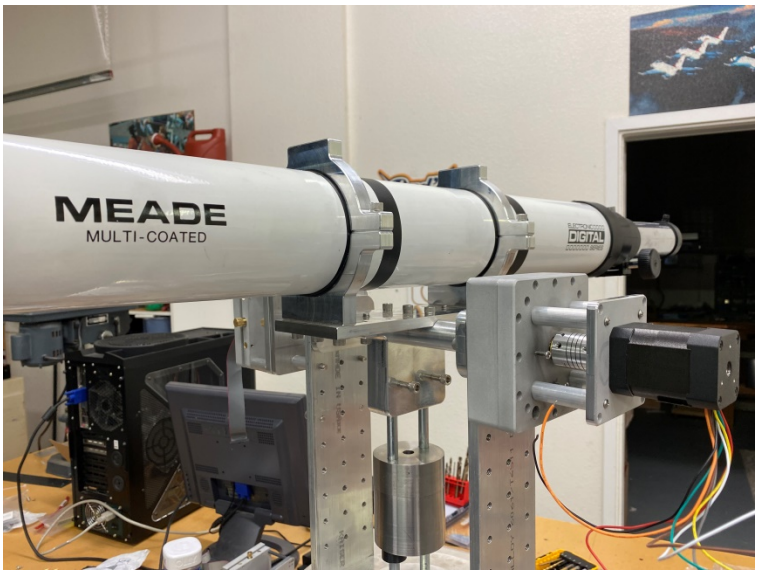


**Figure 15** Fully assembled elevation harmonic drive and associated 3-phase motor

# 3 Raspberry Pi HQ Camera & Telescope Eyepiece

My younger grandkids had difficulty seeing through a conventional telescope eyepiece so I hastened the image capture portion of the project to accommodate them. At present, a Raspberry Pi HQ sensor[3] is used in conjunction with a 40 mm Plossl eyepiece which gives more than enough magnification for my intended purposes. With proper alignment, a full moon just fits within the sensor's field of view.

Spacing between the eyepiece exit pupil and HQ sensor is fairly critical in order to just fill the sensor area. Field curvature is visible in the images which would require attention if I intended to use this telescope for astrophotography purposes. I may integrate a field-flattener into the design at some point in the future, but this is not a driving requirement at this time.

The mating adapter between the telescope focus tube, Plossl eyepiece, and HQ sensor was designed in Fusion360 and then fabricated using an Ender 3D-printer since neither extreme precision nor high strength was required in this portion of the design. The complete assembly is shown in Figure 18


**Figure 16** Focus tube and eyepiece mount adapter portion

[3] *12.3 MP, Sony IMX477 sensor.*

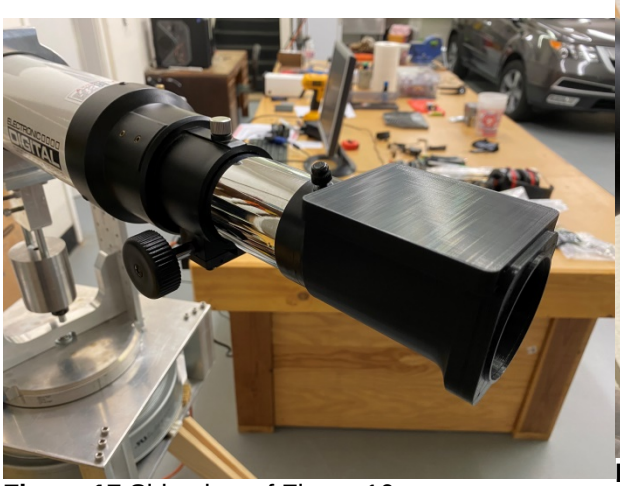

**Figure 17** Side-view of Figure 16

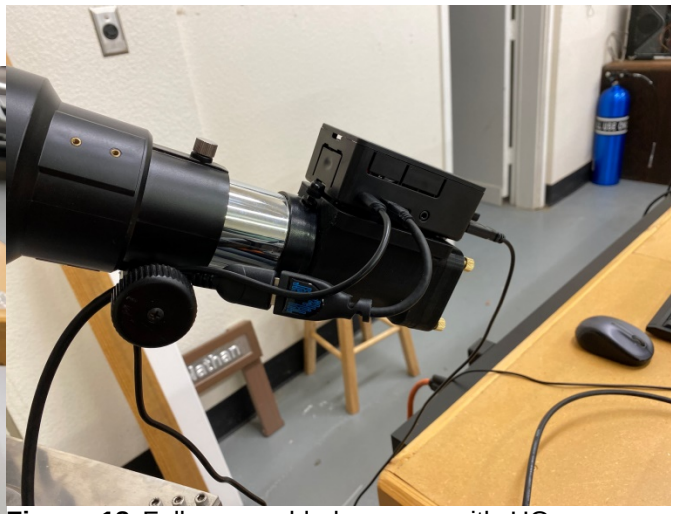

**Figure 18** Fully assembled camera with HQ sensor portion of the adapter now attached, along with a Raspberry Pi4 atop to capture the sensor data. The GPS receiver will be piggy-backed atop the Raspberry Pi processor.

# 4 Finder Scope and Mounting Rings

Once I had the telescope outside my shop looking at terrestrial and astronomical objects, it became immediately clear I needed a lower magnification finder telescope to help with manual observing. I designed and then fabricated the finder telescope around a 50mm achromatic objective lens from an old binocular set.

I then designed and fabricated adjustable mounting rings to hold the telescope along with an adapter plate to mount the finder on the main telescope. These elements along with an extra pair of rings are shown in Figure 19.

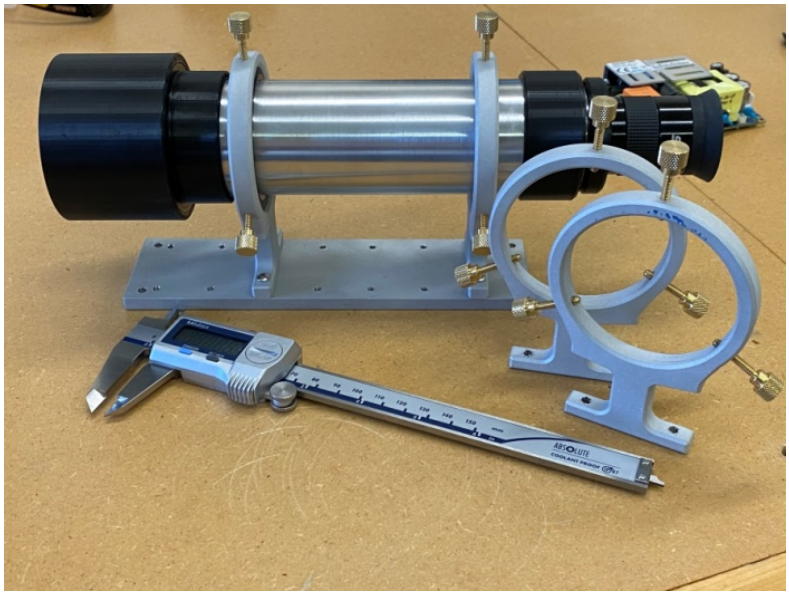

**Figure 19** Finder telescope along with mounting rings and plate adapter

As true with many projects, it is often necessary to fabricate jigs of one sort or another to facilitate a good design. The telescope mounting rings were no exception. The rings were first milled from aluminum plate stock. The fabrication jig shown in Figure 20 and Figure 21 was used to secure each ring in a self-centering vice on the mill's 4th axis for set-screw drilling and tapping. The final complete assembly is shown in Figure 22.

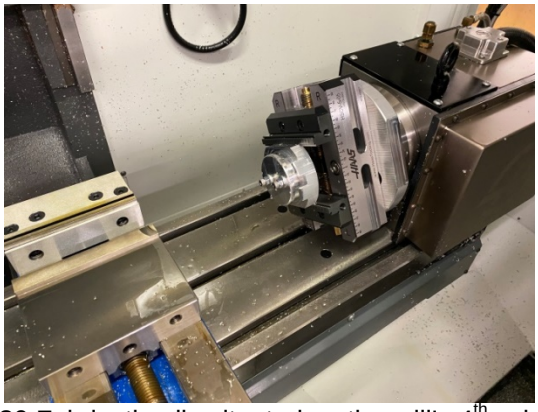

**Figure 20** Fabrication jig situated on the mill's 4th axis

**Figure 21** Close-up of Figure 20

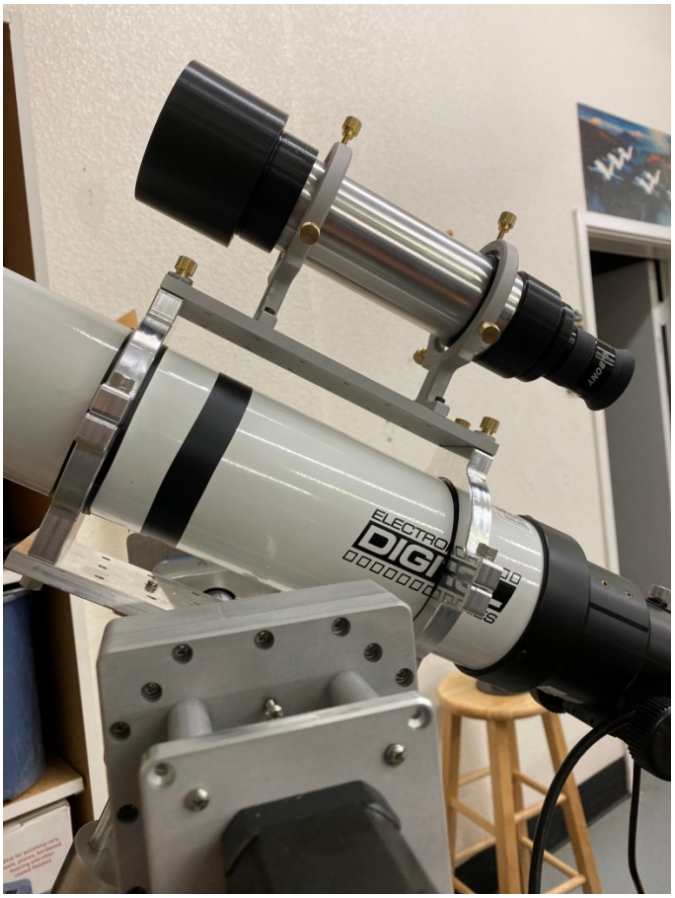

**Figure 22** Finder telescope, ring mounts, and plate adapter installed on the main telescope

# 5 Interim Telescope Tripod

As the desire and need to use the telescope outside my shop developed, I designed and fabricated the tripod mount shown in Figure 23. The circular shelf shown in the figure was milled as were the leg clamps shown in Figure 24. The horizontal tripod struts will be trimmed back and the wood stained and sealed to finish the tripod in the near future.


**Figure 23** Fabricated telescope tripod with the full assembly. A 2 horsepower motor intended for a future telescope mirror grinding machine is shown on the benchtop.

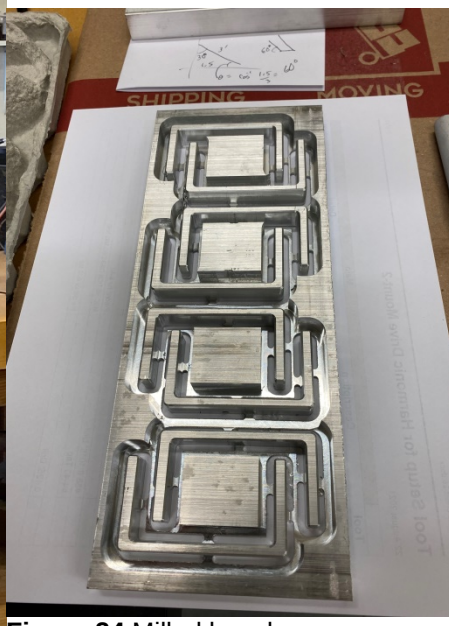


**Figure 24** Milled leg-clamps

# 6 Multi-Processor Configuration

The new multi-processor configuration was shown earlier in Figure 1. This configuration adds additional dimensions to my project to support the broad aspects written about in Part I [7]. High-level details follow.

## *6.1 TI DSP I/O Configuration*

This topic (Figure 25) was first introduced in §2.3 of [1]. The attention at that time was only on supplying higher voltage DC power to the DSP assembly. The following discussion expands the discussion to include (i) expanded digital I/O between the DSP and two DRV8301 Booster Packs, (ii) communication with two optical encoders sharing one SPI port[4], and (iii) hosting a Raspberry Pi4 processor using a second UART port.

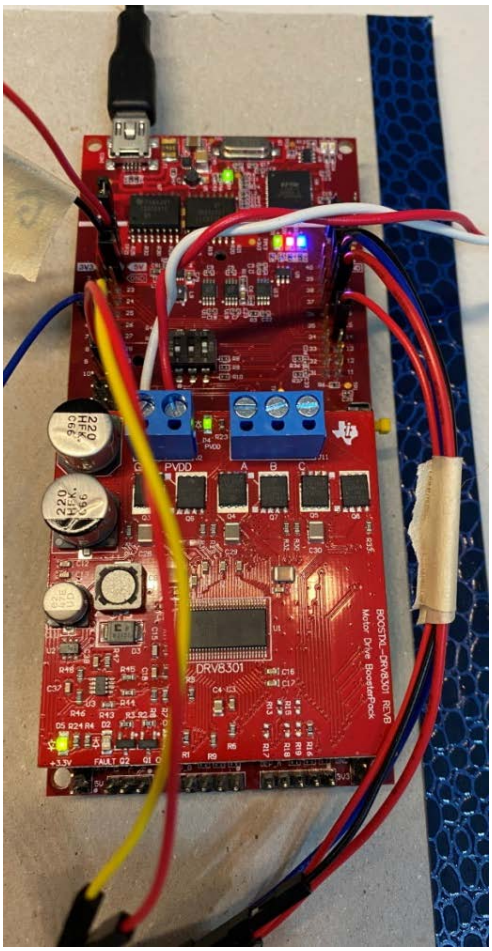


**Figure 25** One DRV8301 Booster Pack installed atop the TMS320F28379D Launchpad XL DSP board. The twisted red-white pair bring in +12V whereas other voltages (3.3V and 5V) are created internally. In the final configuration, the red-white pair will provide 24 VDC to the circuitry.

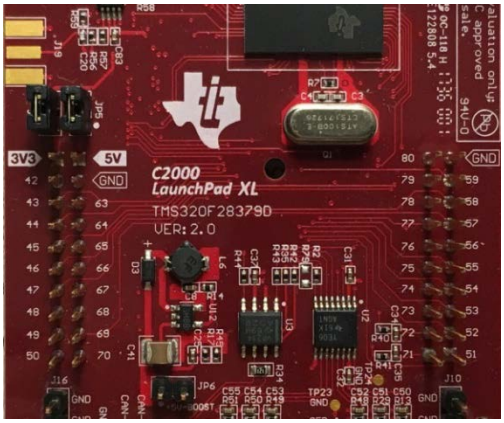


**Figure 26** F28379D Launchpad PWB area furthest from the USB connector, showing from left to right, J5, J7, J8, and J6

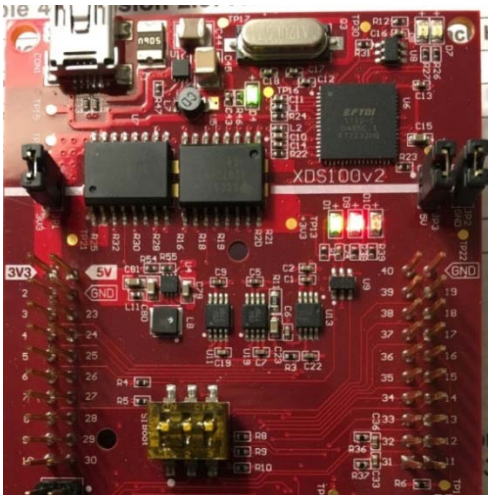


**Figure 27** F28379D Launchpad PWB area closest to the USB connector, showing from left to right, J1, J3, J4, and J2

[4] SPI operation detailed at length in Chapter 18 of [10].

The original DSP ↔ AEAT-9000 Encoder Interface from [1] is provided in Table 6-1 for easy reference. The updated interface needed to support two encoders and two Booster Packs follows later in §6.3.

**Table 6-1** Original DSP ↔ AEAT-9000 Encoder Interface from [1]. Full (I/Q) support is only shown for the azimuth channel. All of the J connectors were used for ease, but some reassignments are needed in order for the TMS320F28379D to be able to support two DRV8301 Booster Packs simultaneously. (None of the elevation encoder signals were actually wired up even though they are shown in the table.)

| DSP Signal Name | DSP GPIO | DSP Pin | Encoder | Encoder Signal Name | Flex-Cable Pin |
|---|---|---|---|---|---|
| SPICLKA | 60 | J1-7 | Az | SCL+ | 28 |
| SPISIMOA | 58 | J2-15 | | | |
| SPISOMIA | 59 | J2-14 | Az | Dout+ | 22 |
| SPISTEA* | 61 | J2-19 | Az | NSL+ | 14 |
| | | | | | |
| ZRSTA | 122 | J2-17 | Az | zero_Reset | 12 |
| nRSTA | 123 | J2-18 | Az | nRST | 26 |
| | | | | | |
| COSINEAP | ADCINA0 | J3-30 | Az | COSINE+ | 1 |
| COSINEAN | ADCINB2 | J3-28 | Az | COSINE- | 2 |
| SINEAP | ADCINC3 | J3-24 | Az | SINE+ | 3 |
| SINEAN | ADCIN14 | J3-23 | Az | SINE- | 4 |
| | | | | | |
| | | | Az | SCL- ($V_{cc}$/2) | 30 |
| | | | Az | NSL- ($V_{cc}$/2) | 16 |
| | | | | | |
| SPICLKB | 65 | J5-47 | El | SCL | |
| SPISIMOB | 63 | J6-55 | | | |
| SPISOMIB | 64 | J6-54 | El | Dout+ | |
| SPISTEB* | 66 | J6-59 | El | NSL | |
| | | | | | |
| ZRSTB | | | El | zero_Reset | |
| nRSTB | | | El | nRST | |
| | | | | | |
| | | | | | |
| +5VA | | J3-21 | | | 27, 29 |
| GNDA | | J3-22 | | | 6, 8, 17, 18 |
| | | | | | |
| +5VB | | J7-61 | | | |
| GNDB | | J7-62 | | | |

## *6.2 DRV8301 Pin-Out*

The pin-out for the DRV8301 is shown in Figure 28. The Booster Pack can be plugged into J1 – J4 of the F28379D LaunchPad XL, or it can be plugged into J5 – J8 of the F28379D LaunchPad XL.

Initial software development work was done with the Booster Pack situated on top of the F28379 opposite the COM port as shown in Figure 25, with the DRV8301's J3 and J4 connectors marrying up with the F28379D's J5-J8 connectors as summarized in Table 6-3. For ease of connection, signal interfacing with the (one) optical encoder was done using LaunchPad XL pins on J1 – J4, but these had to be changed in order for the LaunchPad XL to accommodate two DRV8301 Bosster packs simultaneously.

Ultimately, the Azimuth Booster Pack and associated optical encoder will use only pins in J5 – J8, whereas the Elevation Booster Pack and associated optical encoder will use only pins in J1 – J4 of the F28379D LaunchPad XL.

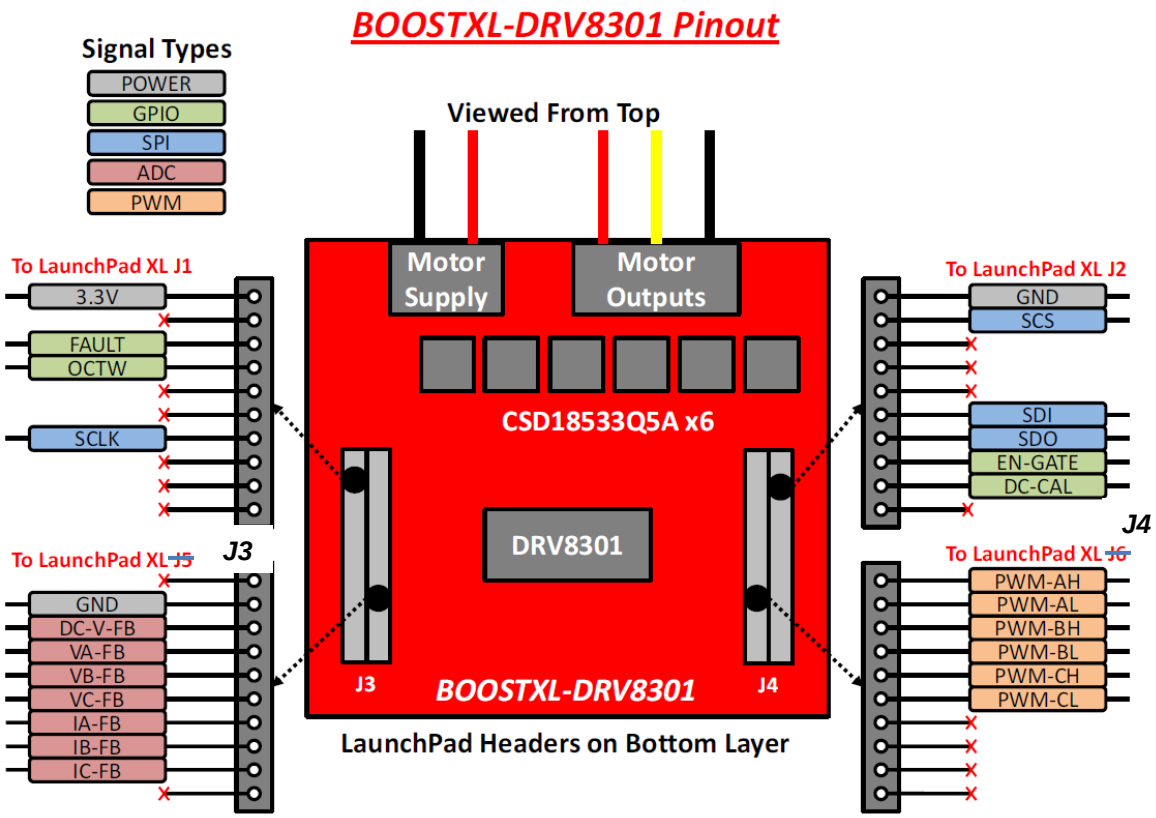


**Figure 28** DRV8301 pin-out [8]

It is helpful to put the DRV8301 Booster Pack, F28379D LaunchPad XL, and optical encoder signals all together in single tables (one for each axis) to see how everything lines up. This has been done in Table 6-2 and Table 6-3.

Four SPI ports are required, two for the two DRV8301 Booster Packs and two for the two optical encoders. However, only 3 SPI ports are available on the F28379D. This can be circumvented as described later in §6.4.

 

**Important Notes [8]:**

1. Resistive scaling on the DRV8301 Booster Pack board is such that only motor voltages up to a ***maximum of 26.3V*** can be used without potentially damaging the F28379D !
2. Current measurement circuitry on the Booster Pack board is scaled for a maximum current of **±16.5A !** Beyond this current range, damage to the F28379D may take place.
3. nFAULT- Fault report indicator
4. nOCTW- Over current / over temperature indicator
5. EN_GATE- Enables gate driver and current shunt amplifiers
6. DC_CAL- When high, shunt amplifier inputs shorted and loads disconnected for DC offset calibration.
7. On-board step-down buck-converter can provide up to 1.5A.

## *6.3 Moving to Two DRV8301 Booster Packs Plus Support for Two Encoders*

The increased capabilities introduced in Figure 1 require more resource planning than the original DSP configuration(s). It is otherwise easy to overlook important functionality and be forced into another redesign.

Each DRV8301 requires a substantial amount of I/O resources from the F28379D so the resource planning begins with accommodating each Booster Pack (Figure 29) and then working with the left-overs to support the remaining functionality.

**>SPI Ports:** Each Booster Pack requires its own SPI port thereby taking up two of the three SPI ports available. The third SPI port is time-multiplexed between the two optical encoders which must be read in a time-synchronous manner.

**>SCI Ports:** One UART port is used to communicate with the host PC. A second SCI port is used to support communication between the F28379D and the Raspberry Pi4. The two other remaining SCI ports are presently unused.

**>Raspberry Pi4:** A video port is provided on the RP4 to support the HQ Camera. The GPS receiver communications is handled with the RP4's SPI port.

**>1PPS:** The GPS receiver's output 1PPS signal is conveyed directly to a GPIO pin of the F28379D to permit accurate time-stamping activities.

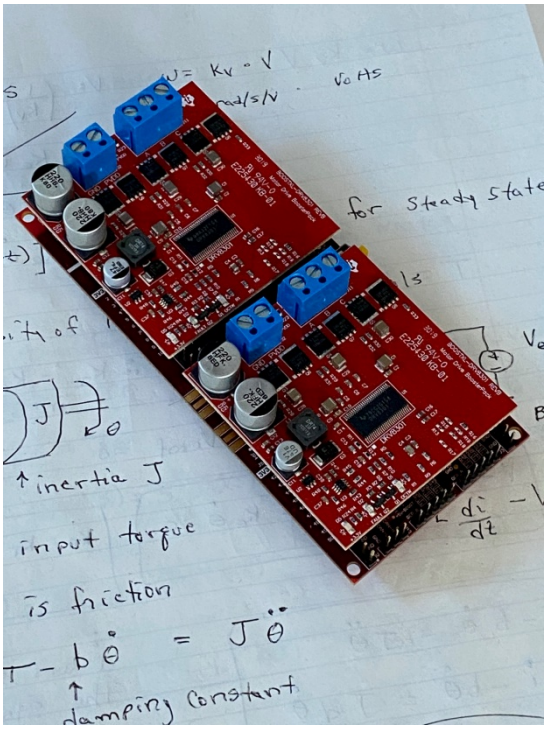


**Figure 29** TMS320F28379D hosting two DRV8301 Booster Packs

**Table 6-2** DRV8301 I/O Mapping[5] to F28379D Configured for Booster Pack Situated Closest to USB Connector for Elevation Axis

| DRV8301 Signal | F28379D Pins | | | | DSP GPIO | F28379D Signal | Non-DRV8301 Usage |
|---|---|---|---|---|---|---|---|
| | J1 | J3 | J4 | J2 | | | |
| 3.3V | 1 | | | | | 3.3V | |
| x | 2 | | | | 32 | GPIO32 / SDAA | LED1 Power |
| FAULT | 3 | | | | 19 | GPIO19 / SCIRXDB / SPISTEA* | |
| OCTW | 4 | | | | 18 | GPIO18 / SCITXDB / SPICLKA | |
| x | 5 | | | | 67 | GPIO67 | nRST_AZ |
| x | 6 | | | | 111 | GPIO111 | nRST_EL |
| SCLK | 7 | | | | 60 | GPIO60 / SPICLKA / SPISIMOB | |
| x | 8 | | | | 22 | GPIO22 / SCITXDB / SPICLKB | Available |
| x | 9 | | | | 105 | GPIO105 / SCLA / SCIRXDD | Available |
| x | 10 | | | | 104 | GPIO104 / SDAA / SCITXDD | Available |
| | | | | | | | |
| x | | 21 | | | | 5V | 5V_ENCODER_B |
| GND | | 22 | | | | GND | GND_ENCODER_B |
| DC-V-FB | | 23 | | | | ADCIN14 / CMPIN4P | |
| VA-FB | | 24 | | | | ADCINC3 / CMPIN6N | |
| VB-FB | | 25 | | | | ADCINB3 / CMPIN3N | |
| VC-FB | | 26 | | | | ADCINA3 / CMPIN1N | |
| IA-FB | | 27 | | | | ADCINC2 / CMPIN6P | |
| IB-FB | | 28 | | | | ADCINB2 / CMPIN3P | |
| IC-FB | | 29 | | | | ADCINA2 / CMP1N1P | |
| x | | 30 | | | | ADCINA0 / DACOUTA | |
| | | | | | | | |
| PWM-AH | | | 40 | | 0 | EPQM1A / GPIO0 | |
| PWM-AL | | | 39 | | 1 | EPWM1B / GPIO1 | |
| PWM-BH | | | 38 | | 2 | EPWM2A / GPIO2 | |
| PWM-BL | | | 37 | | 3 | EPWM2B / GPIO3 | |
| PWM-CH | | | 36 | | 4 | EPWM3A / GPIO4 | |
| PWM-CL | | | 35 | | 5 | EPWM3B / GPIO5 | |
| x | | | 34 | | 24 | OUTPUTXBAR1 / GPIO24 / SPISIMOB | Available- brought out from J4 |
| x | | | 33 | | 16 | OUTPUTXBAR7 / GPIO16 / SPISIMOA | Available- brought out from J4 |
| x | | | 32 | | | DAC1 | |
| x | | | 31 | | | DAC2 | |
| | | | | | | | |
| GND | | | | 20 | | GND | |
| SCS | | | | 19 | 61 | GPIO61 / SPISIMOB / SPISTEA* | |
| x | | | | 18 | 123 | GPIO123 / SD1_C1 / SPISOMIC | Available- brought out from J2 |
| x | | | | 17 | 122 | GPIO122 / SD1_D1 / SPISIMOC | Available- brought out from J2 |
| x | | | | 16 | | RST | |
| SDI | | | | 15 | 58 | GPIO58 / SPISIMOA | |
| SDO | | | | 14 | 59 | GPIO59 / SPISOMIA | |
| EN-GATE | | | | 13 | 124 | GPIO124 / SD1_D2 | |
| DC-CAL | | | | 12 | 125 | GPIO125 / SD1_C2 | |
| x | | | | 11 | 29 | GPIO29 / OUTPUTXBAR6 / SCITXDA | Available- brought out from J2 |

[5] From U26355 LAUNCHXL-F28379D Overview, SPRUI77C, August 2016, March 2019. Not all GPIO etc pins are mapped to J-connectors. See U27334 F28379D Datasheet sprs880k for details.

**Table 6-3** DRV8301 I/O Mapping to F28379D Configured for Booster Pack situated furthest from USB connector

| DRV8301 Signal | F28379D Pins | | | | DSP GPIO | F28379D Signal | Non-DRV8301 Usage |
|---|---|---|---|---|---|---|---|
| | J5 | J7 | J8 | J6 | | | |
| 3.3V | 41 | | | | | 3.3V | |
| x | 42 | | | | 95 | GPIO95 | 1PPS from GPS Receiver |
| FAULT | 43 | | | | 139 | GPIO139 / SCIRXDC | |
| OCTW | 44 | | | | 56 | GPIO56 / SCITXDC | |
| x | 45 | | | | 97 | GPIO97 | SPISOMIC for Optical Encoders |
| x | 46 | | | | 94 | GPIO94 | ~~SPISIMOC for Optical Encoders~~ |
| SCLK | 47 | | | | 65 | GPIO65 / SPICLKB | |
| x | 48 | | | | 52 | GPIO52 / SPICLKC | SPICLKC for Optical Encoders |
| x | 49 | | | | 41 | GPIO41 / SCLB | ZERO_RESET_AZ |
| x | 50 | | | | 40 | GPIO40 / SDAB | ZERO_RESET_EL |
| | | | | | | | |
| x | | 61 | | | | 5V | 5V_ENCODER_A |
| GND | | 62 | | | | GND | GND_ENCODER_A |
| DC-V-FB | | 63 | | | | ADCIN15 / CMPIN4N | |
| VA-FB | | 64 | | | | ADCINC5 / CMPIN5N | |
| VB-FB | | 65 | | | | ADCINB5 | |
| VC-FB | | 66 | | | | ADCINA5 / CMPIN2N | |
| IA-FB | | 67 | | | | ADCINC4 / CMPIN5P | |
| IB-FB | | 68 | | | | ADCINB4 | |
| IC-FB | | 69 | | | | ADCINA4 / CMPIN2P | |
| x | | 70 | | | | ADCINA1 / DACOUTB | |
| | | | | | | | |
| PWM-AH | | | 80 | | 6 | EPWM4A / GPIO6 | |
| PWM-AL | | | 79 | | 7 | EPWM4B / GPIO7 | |
| PWM-BH | | | 78 | | 8 | EPWM5A / GPIO8 | |
| PWM-BL | | | 77 | | 9 | EPWM5B / GPIO9 | |
| PWM-CH | | | 76 | | 10 | EPWM6A / GPIO10 | |
| PWM-CL | | | 75 | | 11 | EPWM6B / GPIO11 | |
| x | | | 74 | | 14 | GPIO14 / OUTPUTXBAR3 / SCITXDB | SCITXDB for RP4 UART |
| x | | | 73 | | 15 | GPIO15 / OUTPUTXBAR4 / SCIRXDB | SCIRXDB for RP4 UART |
| x | | | 72 | | | DAC3 | |
| x | | | 71 | | | DAC4 | |
| | | | | | | | |
| GND | | | | 60 | | GND | |
| SCS | | | | 59 | 66 | GPIO66 / SPISTEB* | |
| x | | | | 58 | 131 | GPIO131 / SD2_C1 | READAZ* |
| x | | | | 57 | 130 | GPIO130 / SD2_D1 | READEL* |
| x | | | | 56 | | RST | |
| SDI | | | | 55 | 63 | SPISIMOB / GPIO63 | |
| SDO | | | | 54 | 64 | SPISOMIB / GPIO64 | |
| EN-GATE | | | | 53 | 26 | GPIO26 / SD2_D2 | |
| DC-CAL | | | | 52 | 27 | GPIO27 / SD2_C2 | |
| x | | | | 51 | 25 | GPIO25 / OUTPUTXBAR2 / SPISOMIB | Available |

## *6.4 Using GPIO to Emulate SPISTE**

The F28379D has only 3 SPI ports, two of which are dedicated to the DRV8301 Booster Packs as mentioned earlier. Fortunately, individual GPIO pins can be configured to mimic SPISTE* signals to as many slave devices as desired with all slaves sharing the SPICLK, SPISIMO, and SPISOMI lines[6]. GPIO131 will perform the SPISTE* function for the azimuth encoder read operation, and GPIO130 will perform the same function for the elevation encoder read function.

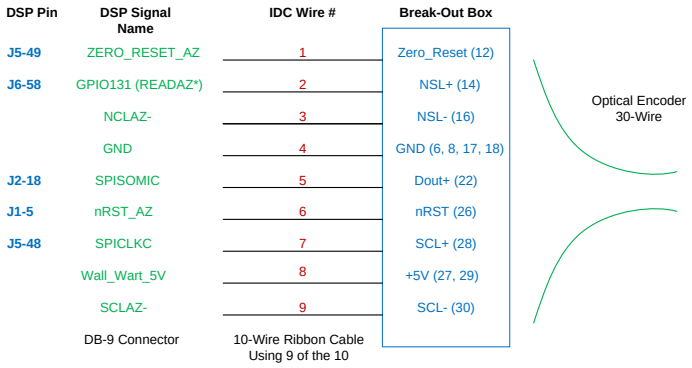


**Figure 30** DSP / ribbon cable / encoder wiring[7] for Az optical encoder. Duplicate wiring diagram for El optical encoder except GPIO130 is used for READEL*, ZERO_RESET_AZ and nRST_AZ similarly modified. The red ribbon cable wire is taken to be #1.

# 7 References[8]


1. J.A. Crawford, “Photogrammetry for Non-Invasive Terrestrial Position/Velocity Measurement of High-Flying Aircraft- Part V: Optical Encoder Completion and Transition to TI DSP,” January 2021, U24933.
2. _____, “Photogrammetry for Non-Invasive Terrestrial Position/Velocity Measurement of High-Flying Aircraft- Part 6A,” January 2022.
3. _____, “Photogrammetry for Non-Invasive Terrestrial Position/Velocity Measurement of High-Flying Aircraft- Part V,” U24933, January 2021.
4. _____, “Photogrammetry for Non-Invasive Terrestrial Position/Velocity Measurement of High-Flying Aircraft- Part IV: Optical Encoder and Elevation Drive,” U24933, August 2020.
5. _____, “Photogrammetry for Non-Invasive Terrestrial Position/Velocity Measurement of High-Flying Aircraft- Part III,” U24933, November 2019.
6. _____, “Photogrammetry for Non-Invasive Terrestrial Position/Velocity Measurement of High-Flying Aircraft- Part II, Direct-Drive Motors,” U24933, April 2019.
7. _____, “Photogrammetry for Non-Invasive Terrestrial Position/Velocity Measurement of High-Flying Aircraft- Part I,” 2017, U24866.
8. Texas Instruments, “BOOSTAXL-DRV8301 Hardware User’s Guide,” SLVU974, Oct. 2013, U27331.
9. _____, “The TMS320F2837xD Architecture: Achieving a new Level of High Performance,” SPRT720 Feb 2016, U27333.
10. _____, “TMS320F2837xD Dual-Core Delfino Microcontrollers, Technical Reference Manual,” Jan. 2019, SPRUHM8H, U26369.
11. Elliott D. Kaplan and Christopher J. Hegarty, *Understanding GPS, Principles and Applications*, 2nd ed., Artech House, 2006.


[6] Described in §18.2.1 of [10].
[7] From U27771_Part6_Figures.vsd. Revised here in Part B compared to Part A.
[8] Expanded section from Part A.

 

12. Harmonic Drive LLC, "HarmonicDrive® Reducer Catalong", U27619.
13. Oliver Montenbruck and Thomas Pfleger, *Astronomy on the Personal Computer*, 1999, Springer-Verlag.
14. Eric Burgess, *Celestial Basic, Astronomy on Your Computer*, 1985, Sybex.
15. Jean Meeus, *Astronomical Algorithms*, 2nd ed., 1998, William-Bell.
16. Roger R. Bate, Donald D. Mueller, and Jerry E. White, *Fundamentals of Astrodynamics*, Dover, 1971.
17. Pratap Misra and Per Enge, *Global Positioning System, Signals, Measurements, and Performance*, 2nd ed., Ganga-Jamuna Press, 2012.
18. Richard H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*, Revised Ed., American Institute of Aeronautics and Astronautics, 1999.

**Table 7-1** Reference Materials

| Title | # | Comments |
|---|---|---|
| TMS320C28x Optimizing C/C++ Compiler v20.8.0.STS, U27392. | U27392 | Compiling, linking, optimization, pragma directives |
| TMS320F2837xD Dual-Core Delfino Microcontrollers, Technical Reference Manual, U26369. | U26369 | Interrupts, timers, memory, DMA, CLA overview & building application** |
| Sensorless Field Oriented Control of 3-Phase Induction Motors Using Control Law Accelerator (CLA) | U27328 | Digital motor control on CLA** |
| C2000 CLA Software Development Guide | U27590 | CLA introduction |
| Programming TMS320x28xx and TMS320x28xxx Peripherals in C/C++, U26359. | U26359 | |
| F2837xD Firmware Development Package, U27306. | U27306 | |
| Sensored Field Oriented Control of 3-Phase Permanent Magnet Synchronous Motors Using F2837x, U27432. | U27432 | |
| Sensored Field Oriented Control of 3-Phase Permanent Magnetic Synchronous Motors Using TMS320F2837x, U27302, U27322. | U27302<br>U27322 | |
| Dual-Axis Motor Control Using FCL and SFRA on a Single C2000 MCU | U27427 | |

# 8 Appendix: Harmonic Drive Assembly

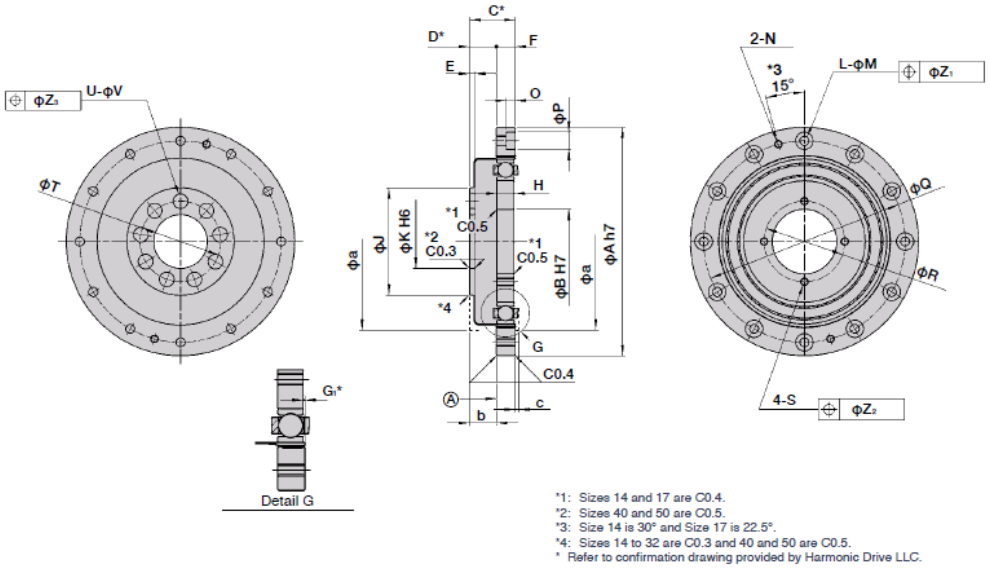


**Figure 31** Harmonic drive dimensions from [12]

**Table 8-1** Dimension Excerpt from [12]

| Symbol \ Size | | | 25 |
|---|---|---|---|
| ΦA h7 | | | 85 $^{0}_{-0.035}$ |
| ΦB H7 | | | 24 $^{+0.021}_{0}$ |
| C* | | | 17 |
| D* | | | 10 $^{+0.3}_{0}$ |
| E | | | 2 |
| F | | | 7 |
| $G_1$* | | | 0.4 $^{+0.2}_{0}$ |
| H | | | 6.3 $^{0}_{-0.1}$ |
| ΦJ | | | 40 |
| ΦK H6 | | Standard | 20 $^{+0.013}_{0}$ |
| | | BB spec. | 24 $^{+0.013}_{0}$ |
| L | | | 12 |
| ΦM | | | 3.4 |
| N | | | M3 |
| O | | | 3.3 |
| ΦP | | | 6.5 |
| ΦQ | | | 75 |
| ΦR | | | 30 |
| S | | | M3 |
| ΦT | | Standard | 30 |
| | | BB spec. | 32 |
| U | | Standard | 9 |
| | | BB spec. | 12 |
| ΦV | | Standard | 5.5 |
| | | BB spec. | 4.5 |
| $ΦZ_1$ | | | 0.2 |
| $ΦZ_2$ | | | 0.2 |
| $ΦZ_3$ | | Standard | 0.25 |
| | | BB spec. | 0.25 |
| Minimum housing clearance | Φa | | 66 |
| | b | | 10 |
| | c | | 1.5 |
| Mass (kg) | | | 0.24 |

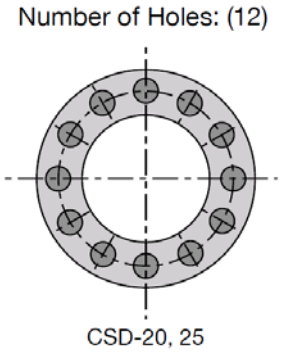


**Figure 32** Mounting holes

# 9 Appendix: Azimuth Drive- Bottom Plate for Encoder Mount

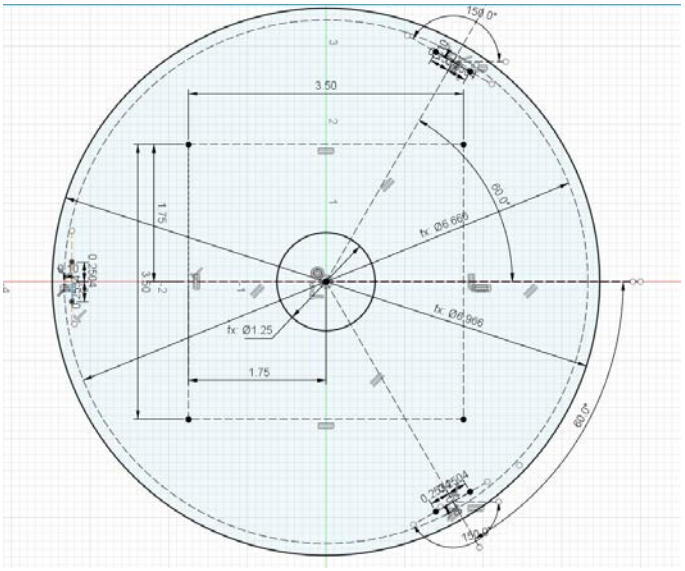


**Figure 33** Telescope Az Encoder Mounting Plate

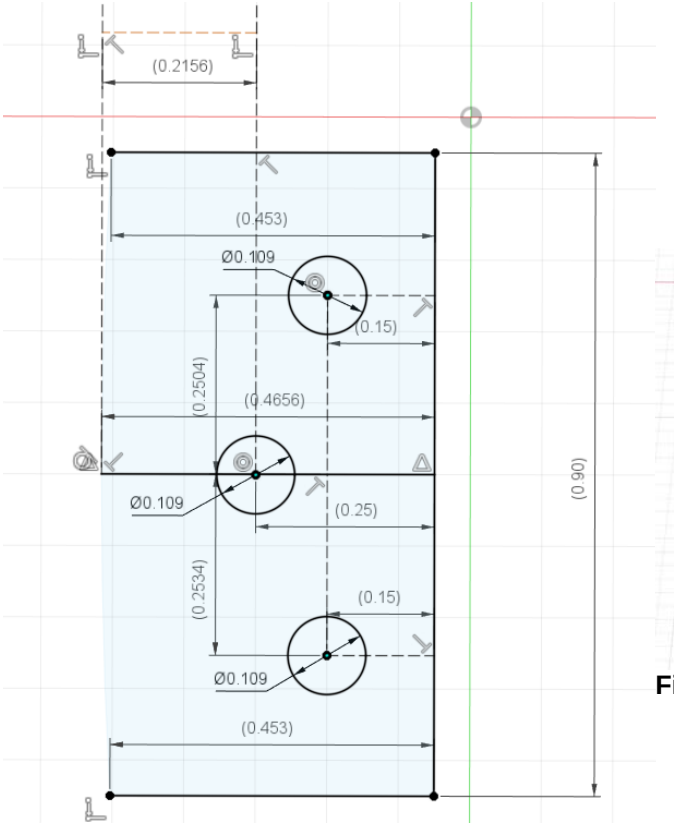


**Figure 34** Detail for mounting hole region of “feet”

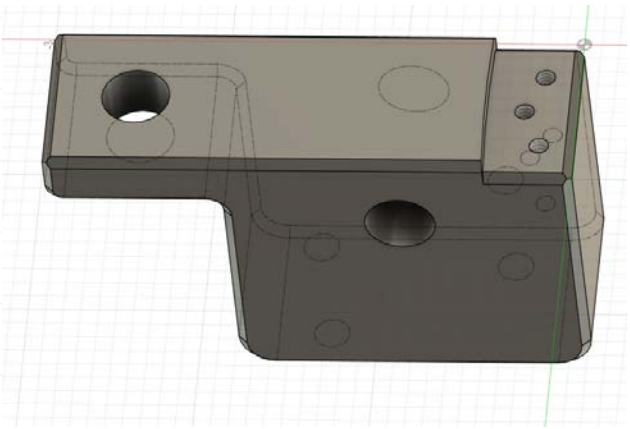

**Figure 35** One of the “feet”

 

# 10 Appendix: Telescope Optics and CCD Sensor

I have tentatively chosen the Raspberry Pi 4 (RP4) microcomputer working in conjunction with a Raspberry Pi HQ CCD camera for the project. Since my end objective is expressly *not* astronomical in nature at this time, I did not want to spend a lot of money on a good imaging-CCD which can easily run $1,000 or more new. I also have some side-pursuits which would benefit from a more reasonably priced CCD sensor.

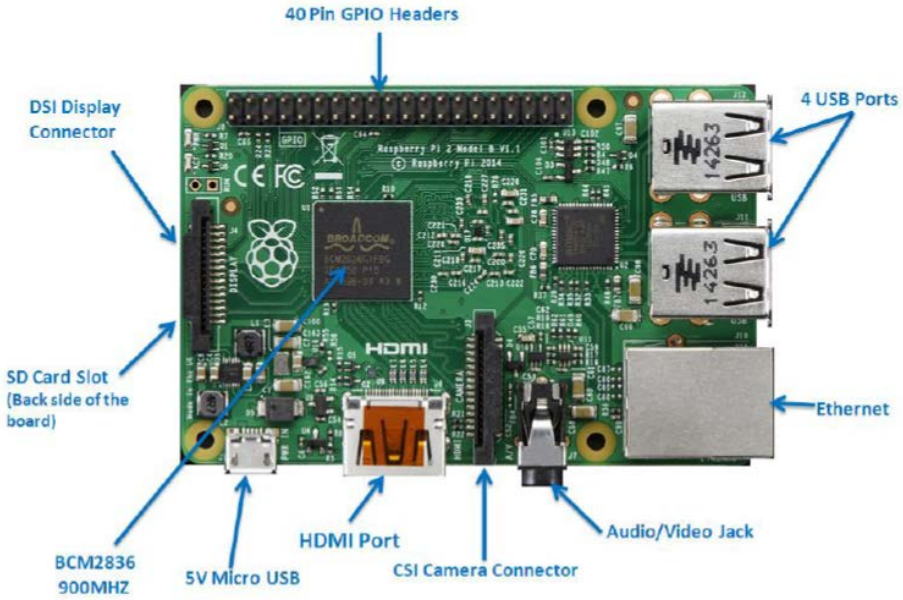


**Figure 36** Raspberry Pi 4[9]

**Figure 37** Front view of the HQ sensor[10]

The HQ RP4 sensor's details include:

- 12.3 Megapixels ( 3040 x 4056 )
- Pixel Size: 1.55 μm x 1.55 μm
- 7.9 mm diagonal
- 0.2475"x 0.1855"

Sizing the optical train to the sensor is very important if good results are to be obtained.

## *10.1 CCD Sensor at Prime Focus*

One of the most simple optical arrangements is to use an objective telescope lens with the CCD sensor placed at the prime focus as shown in Figure 38. The thin lens formula can be used to closely estimate image size, magnification, etc. and is given by

[9] From U27684, Raspberry Pi Tutorial.
[10] From "HQ Camera Product Brief,"U27679.

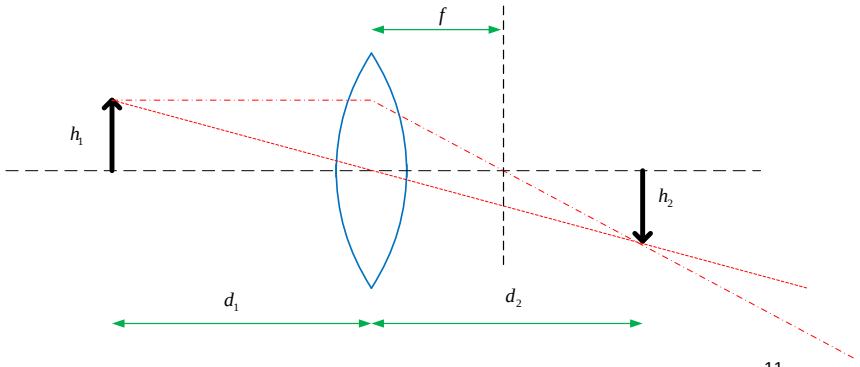


**Figure 38** Simple arrangement using objective lens and CCD sensor at the focus[11]

$$\frac{1}{f} = \frac{1}{d_1} + \frac{1}{d_2} \tag{1}$$

For distant objects,

$$\theta_1 = \sin^{-1}\left(\frac{h_1}{d_1}\right) \cong \frac{h_1}{d_1} \tag{2}$$~

In the full-moon case, the extended angle is about 0.50°. Rearranging (1) produces

$$d_2 = \frac{f d_1}{d_1 - f} \tag{3}$$

with the magnification given by

$$M = \frac{d_2}{d_1} \tag{4}$$

Normally the sign of each distance is carried along in the calculations to determine where the images are formed, but the outcome is already known in this simple case. Since it is also true

$$M = \frac{h_2}{h_1} \tag{5}$$

the image height can be solved for as

$$\begin{aligned} h_2 &= \left(\frac{f}{d_1 - f}\right) h_1 = \frac{f}{d_1\left(1 - \frac{f}{d_1}\right)} h_1 \cong \frac{f}{1 - \frac{f}{d_1}} \theta_1 \\ &\cong f\left(1 + \frac{f}{d_1}\right)\theta_1 \end{aligned} \tag{6}$$

where $\theta_1$ is in radians. Since $d_1 >> f$ , the image height is simply given by

[11] U27712 CCD Telescope Optics.vsd.

 

$$h_2 = f\theta_1 \tag{7}$$

Several values are computed in Table 10-1, illustrating how the CCD sensor size is more than filled for focal lengths greater than roughly 12".

**Table 10-1** Moon Image Height vs Objective Focal Length per (7)

| Focal Length, in | Image Height, in |
|---|---|
| 12 | 0.105 |
| 24 | 0.209 |
| 36 | 0.314 |
| 48 | 0.419 |

The resolving power of the telescope is dictated by the diameter of the objective lens as closely approximated by

$$\varphi_{resolve} \cong \frac{4.5}{D_{obj}} \quad arc\text{-}sec \tag{8}$$

when the objective's diameter $D_{obj}$ is given in inches. Ideally, the resolving power will be approximately the size of one CCD sensor's pixels.

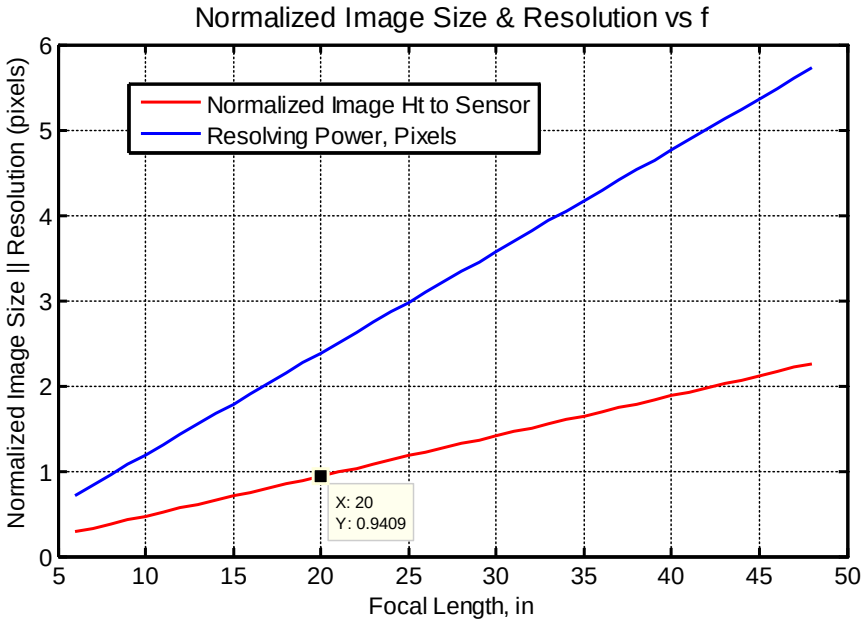


**Figure 39** Resolving power and (moon) image size versus focal length[12]

$$f_{eff} = \frac{f_1 f_2}{f_1 + f_2 - d} \tag{9}$$

$$f_2 = \frac{f_{eff}\left(d - f_1\right)}{f_{eff} - f_1} \tag{10}$$

[12] u27713_ccd_optics.m.

# 11 Appendix: Raspberry Pi4 External Interface

**Table 11-1** Raspberry Pi4 External Interface Pin-Out

| Function | Pin Number | Pin Number | Function |
|---|---|---|---|
| 3V3 | 1 | 2 | 5V |
| SPI3 MOSI/SDA3 | 3 | 4 | 5V |
| SPI3 SCLK/SCL3 | 5 | 6 | GND |
| SPI4 CE0 N/SDA 3 | 7 | 8 | TXD1/SPI5 MOSI |
| GND | 9 | 10 | RXD1/SPI5 SCLK |
| | 11 | 12 | SPI6 CEO N |
| SPI6 CE1 N | 13 | 14 | GND |
| SDA6 | 15 | 16 | SCL6 |
| 3V3 | 17 | 18 | SPI3 CE1 N |
| SDA5 | 19 | 20 | GND |
| RXD4/SCL4 | 21 | 22 | SPI4 CE1 N |
| SCL5 | 23 | 24 | SDA4/TXD4 |
| GND | 25 | 26 | SCL4/SPI4 SCLK |
| SPI3 CE0 N/TXD2/SDA6 | 27 | 28 | SPI3 MISO/SCL6/RXD2 |
| SPI4 MISO/RXD3/SCL3 | 29 | 30 | GND |
| SPI4 MOSI/SDA4 | 31 | 32 | SDA5/SPI5 CEO N/TXD5 |
| SPI5 MISO/RXD5/SCL5 | 33 | 34 | GND |
| SPI6 MISO | 35 | 36 | SPI1 CE2 N |
| SPI5 CE1 N | 37 | 38 | SPI6 MOSI |
| GND | 39 | 40 | SPI6 SCLK |
| | | | |
| I2C | | | Ground |
| UART | | | 5V Power |
| SPI | | | 3V3 Power |

*---------- Part B Materials Follow ----------*

In hindsight, I should have dispensed with “Part B” and just moved on to Part 7. Hopefully this material will still flow in a sensible manner.

# 12 Main Electronics Package

The TMS320 Launchpad board, power supply, and connectors clearly needed a home so I designed and fabricated the assembly taking place in Figure 40. I wanted a clean design with all internal signals connectorized. Even with ribbon cables, the number of connections involved became rather busy as shown in Figure 51.

The chassis front panel (which was milled, anodized, and then engraved) is shown in Figure 40 and most of the internal constituent electronic ingredients are shown immediately before chassis integration in Figure 41. The final assembly was quite a bit more complicated though as shown in Figure 51. The piggy-back board shown in the lower portion of Figure 41 was designed to plug directly into the back-side of the TMS320 Launchpad board and the associated schematic is shown in Figure 45. The piggy-back board was designed to perform the following tasks:

1. Serve as a convenient signal-routing conduit for signals to/from the DSP and the Az/El optical encoders
2. Host a triac AC Power On/Off circuit to avoid any user direct contact with the 120VAC input supply voltage
3. Create $V_{cc}/2$ bias voltages for the optical encoder digital interfaces in order to mimic differential signals.
4. Conveniently route DSP signals to front-panel LEDs for the user interface

More specifically, the chassis hosts:

- TMS320F28379D LaunchPad & Two DRV8301 Booster Packs (§**Error! Reference source not found.**)
- AC to 24 VDC 200W power supply[13], Astrodyne TDI ASM201-240-BNH-PF1 (§13.1)
- Interface (piggy back) board
- DB-9 connectors for optical encoder support (2)
- DB-9 connectors for Az & El motor support (2)
- SPI port to external Raspberry Pi4 (+ GNSS receiver)
- Cannibalized wall-wart USB 5V power supply
- Electronics cooling fan
- Triac circuit for 110VAC control

All of the electrical interfaces of the integrated chassis are described in §6. The Raspberry Pi4 + HQ Camera arrangement is described separately in §10 and §11.

Based upon Table 6-2 and Table 6-3, J2, J4, J5, and J6, the signals needed for encoder support are shown below in Red whereas the remaining GPIOs which are still available (after the Booster Pack needs are accounted for) are shown in Green in the lists below. Signals in black are already dedicated to the DRV-8301 driver packs. See Figure 43 for a signal-placement view.

**J1:** P104 | P105 | P22 | P60 | P111 | P67 | P18 | P19 | P32 | 3V3
**J2:** P29 | P125 | P124 | P59 | P58 | RST | P122 | P123 | P61 | GND
**J3:** ADC | ADC | ADC | ADC | ADC | ADC | ADC | ADC | GND | 5V
**J4:** DAC2 | DAC1 | P16 | P24 | P05 | P04 | P03 | P02 | P01 | P00
**J5:** P40 | P41 | P52 | P65 | P94 | P97 | P56 | P139 | P95 | 3V3
**J6:** P25 | P27 | P26 | P64 | P63 | RST | P130 | P131 | P66 | GND
**J7:** ADC | ADC | ADC | ADC | ADC | ADC | ADC | ADC | GND | 5V
**J8:** DAC4 | DAC3 | P15 | P14 | P11 | P10 | P9 | P8 | P7 | P6

[13] U27715.

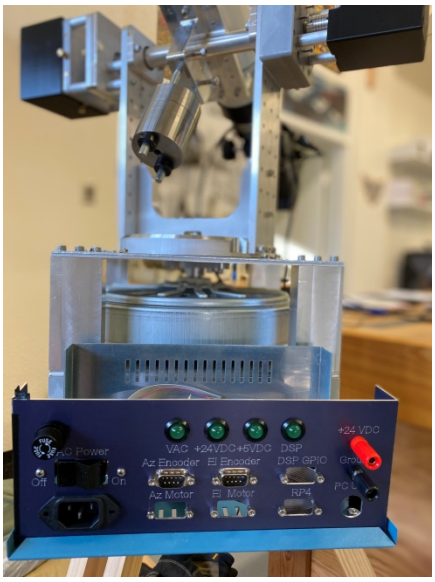


**Figure 40** Front panel of electronics chassis while under development

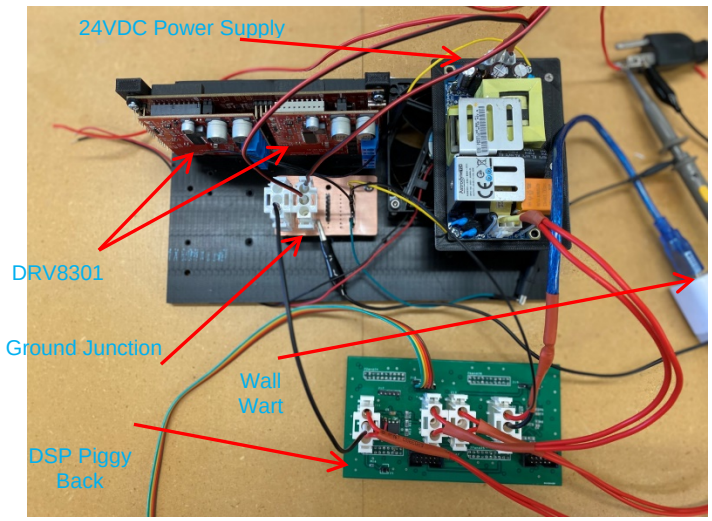


**Figure 41** Chassis components under development, just before integration. The bottom black board is a sheet of ¼" thick Delrin plastic.

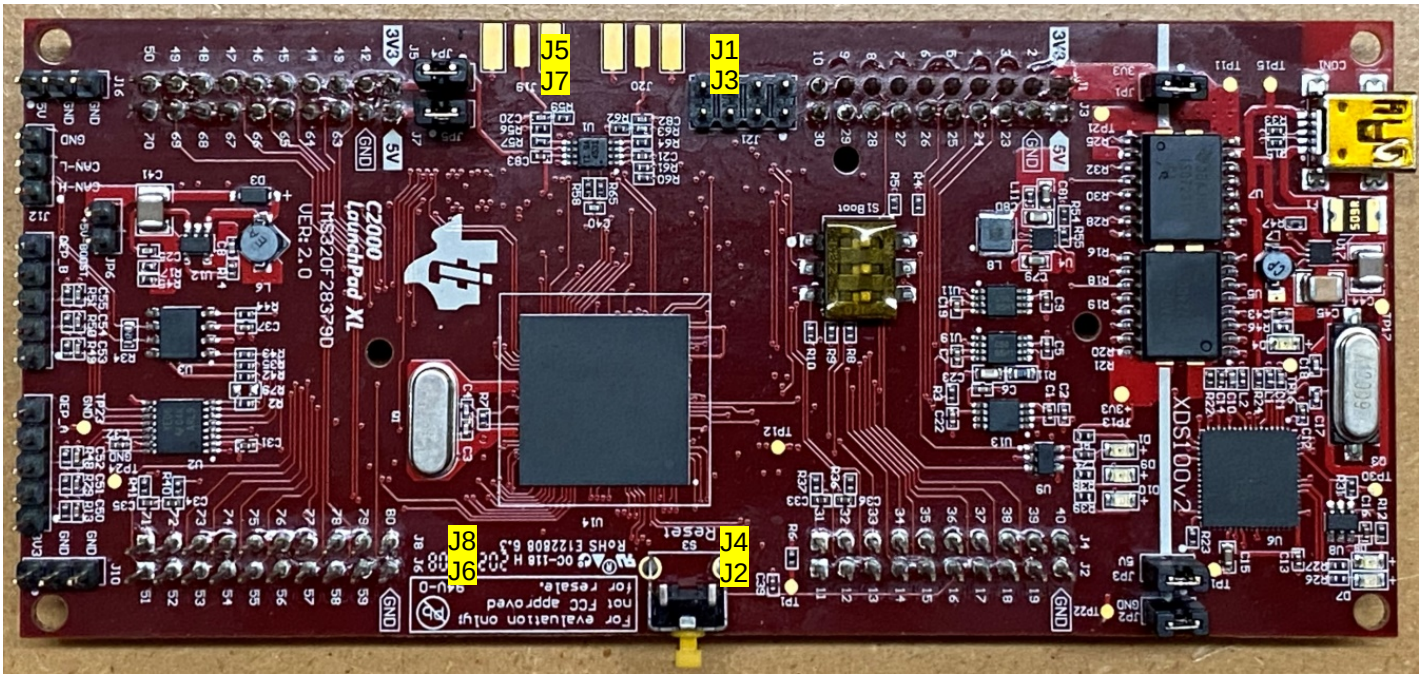


**Figure 42** Top-side of the TMS320F28379D Launchpad board. Note the yellow reset button at the bottom.

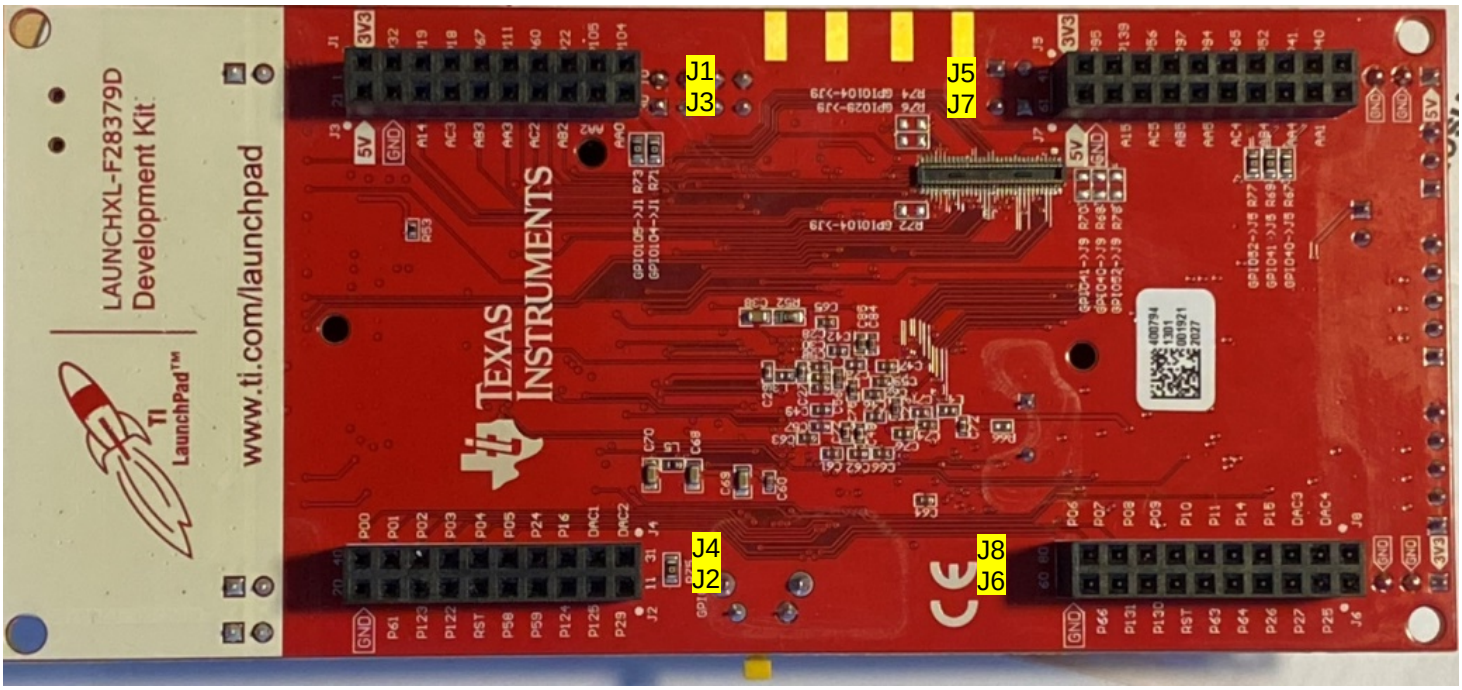


**Figure 43** Back-side of the TMS320F28379D Launchpad board. Note the yellow reset button at the top.

Most of the schematic is involved with routing DSP signal pins to headers thereby making connections to the external encoders and Raspberry Pi simple and clean. I did, however, add a 120 VAC triac circuit for user protection. I reasoned that I may use the telescope while it is located outside on damp grass and I did not like the idea of switching the full 120 VAC with a simple mechanical switch. The triac circuit involves only switching 5 VDC at a few mA of current. Another perspective of the triac circuit involved is shown in Figure 46.

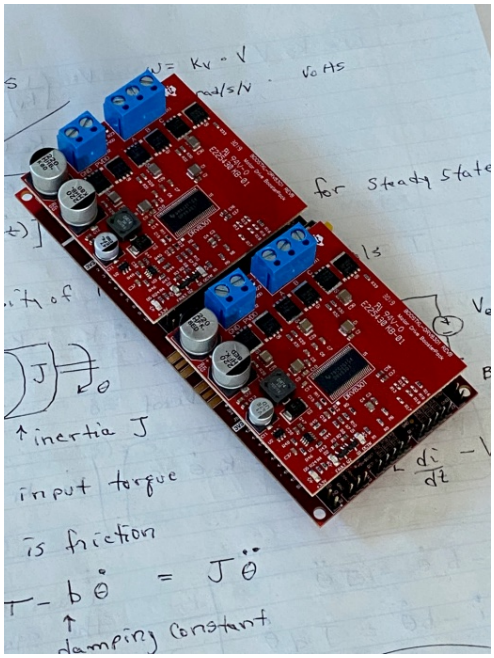


**Figure 44** TMS320F28379D LaunchPad with Dual DRV8301 Booster Packs installed

 

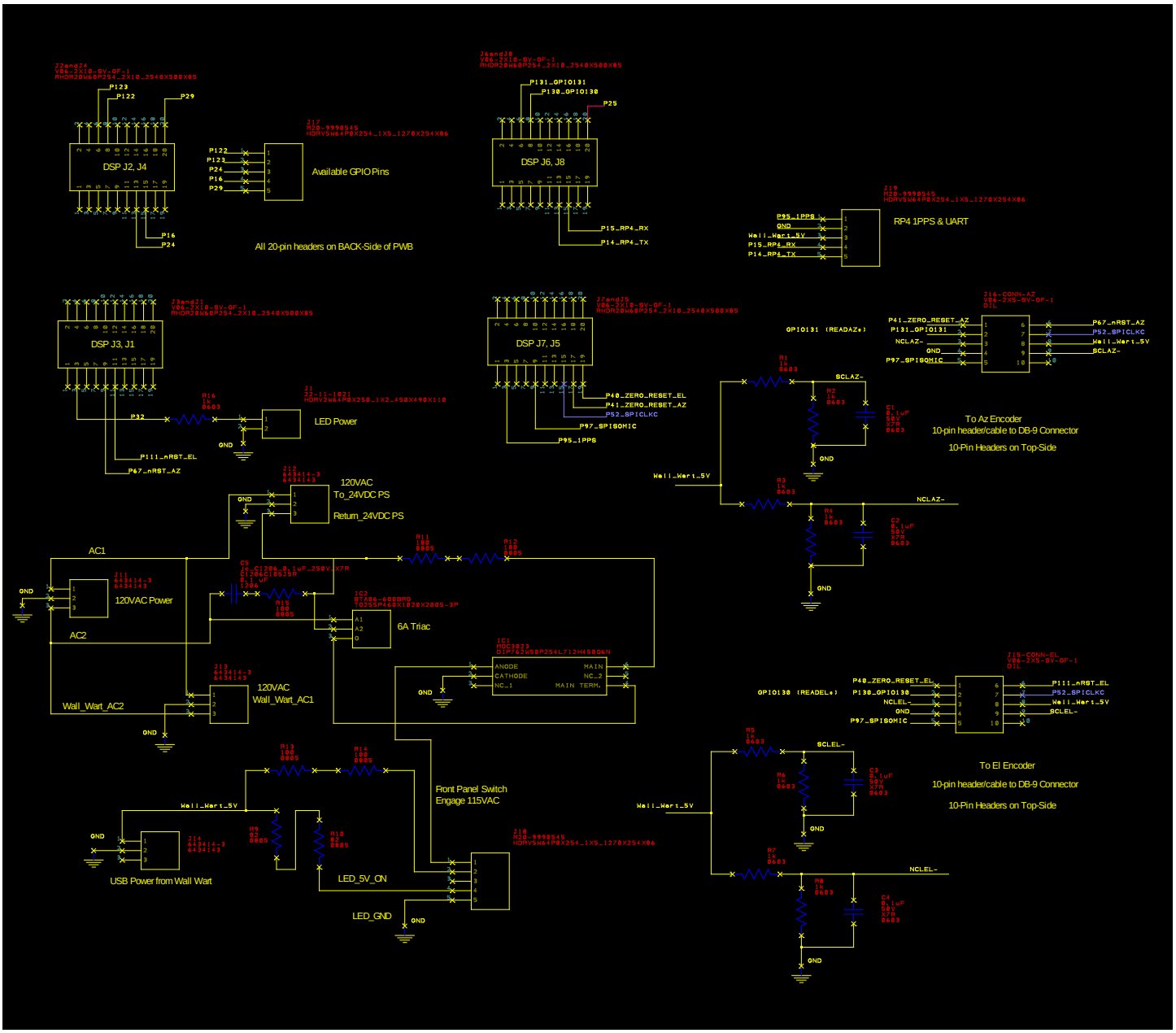


**Figure 45** Schematic for piggy-back DSP board

 

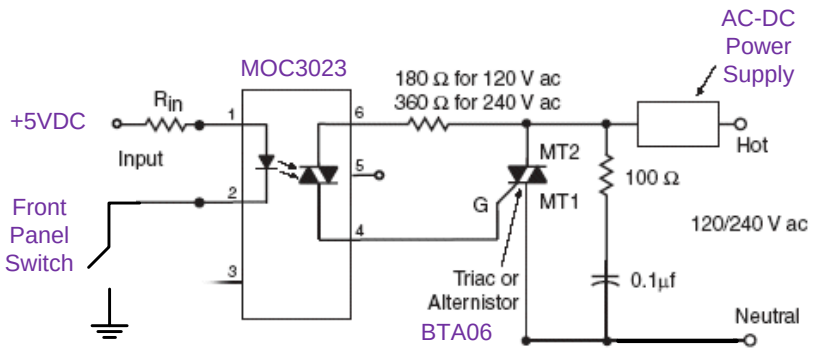


**Figure 46** Additional perspective[14] for the triac circuit used in Figure 45

The AC-DC switching power supply referred to in the schematics is an Astrodyne TDI ASM201-240-BNH-PF1. Its 24 VDC output was connected to an incandescent light for testing purposes. The rise-time and fall-time behaviors are shown in Figure 47 and Figure 48.

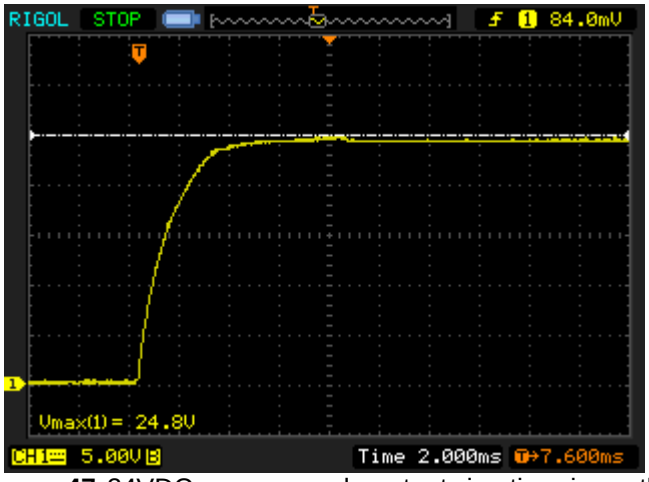


**Figure 47** 24VDC power supply output rise-time is on the order of 4 msec

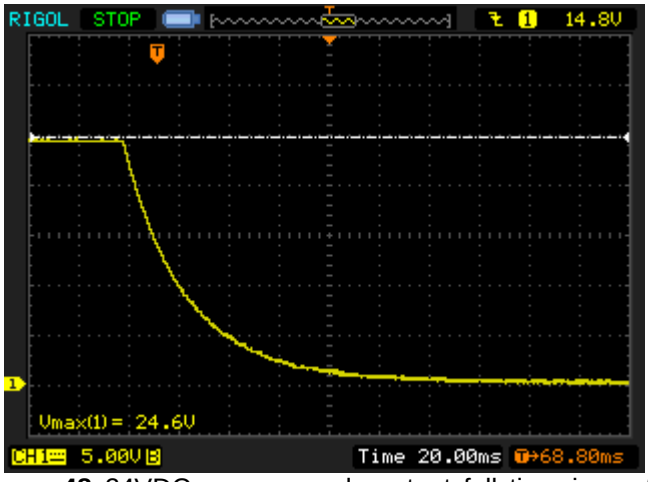


**Figure 48** 24VDC power supply output fall time is on the order of 50 msec

[14] U28353_Triac_Circuit.vsd.

Using two DRV8301 motor drivers with the TMS320F28379D Launchpad requires some attention to board-level jumpers so that DC power creation is done without conflict. More specifically:

- JP1, JP2, JP3 should not be connected so that the USB's 5V supply is not used.
- JP4, JP5 should not be connected because this would result in both DRV8301 Booster Packs sourcing 3.3V into the Launchpad which could be problematic. With these two jumpers disconnected, the Booster Pack sitting atop J1-J4 provides 3.3V to everything, including the second Booster Pack
- JP6 should be connected. This enables the Launchpad's DC-DC converter to create the needed 5V supply from the 3.3V supply.

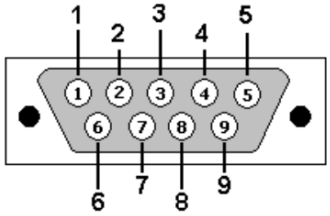


**Figure 49** Looking into the DB-9 male connector[15]

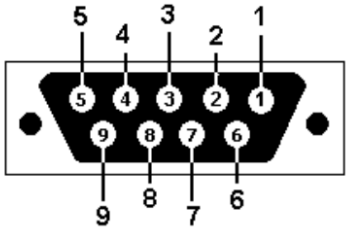


**Figure 50** Looking into the DB-9 female connector[16]

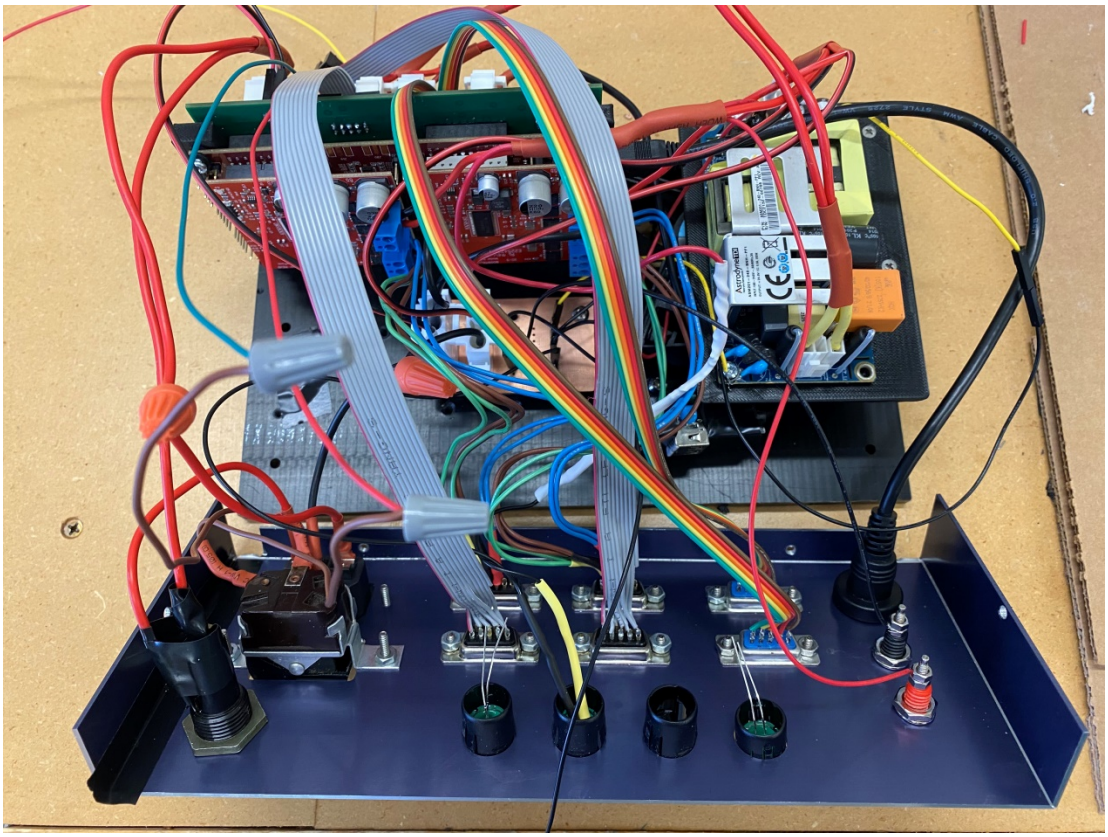

**Figure 51** Internal circuitry and wiring just before buttoning up the electronics package

[15] U27999 https://www.pbxdom.com/wp-content/uploads/2015/09/1432204807_rs232_db9.gif
[16] U27999 https://www.pbxdom.com/wp-content/uploads/2015/09/1432204827_rs232_db9_f.gif

 

## *12.1 Front-Panel Anodizing*

As already mentioned, the front-panel for the electronics package shown in Figure 40 and Figure 51 was anodized (6061) aluminum. I have wanted to have my own anodizing setup for quite some time because the process makes aluminum surface so much more scratch resistant than raw material even if coloring is not added. My anodizing setup is the subject of a separate memo, however, and I have only included a few incidental photos of it in this material. The etching was done using my Haas mill and a diamond-tipped cutter.

The main anodizing tank is shown in Figure 53. Electrical connections for the anodizing current are partially shown, along with fish tank bubbler hoses and a connection to the internal temperature sensor. At the back edge of the tank's top are inlet and outlet surgical tubing hoses which circulate ice-cold water through a cooling cool immersed in the bottom of the tank. A circulating pump combined with the temperature sensor and an external controller keep the anodizing solution at precisely 72$^{o}$ F.

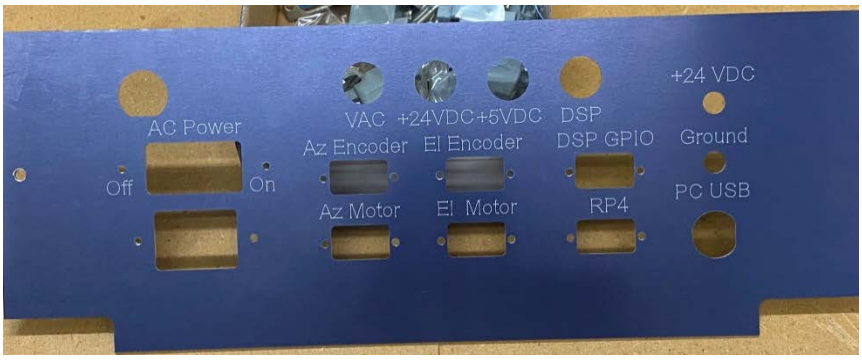


**Figure 52** Front panel after milling, anodizing, and etching

**Figure 53** Main anodizing container. Degreaser and acid neutralizing baths are shown on opposite sides.

# 13 Miscellaneous

## *13.1 AC-DC (24V) Switching Power Supply*

The internal AC-DC power supply used within the electronics package is an Astrodyne TDI ASM201-240-BNH-PF1 115 VAC, 24 VDC, 8.4A, 92% Efficiency. This can be purchased directly from Astrodyne or through Digikey.

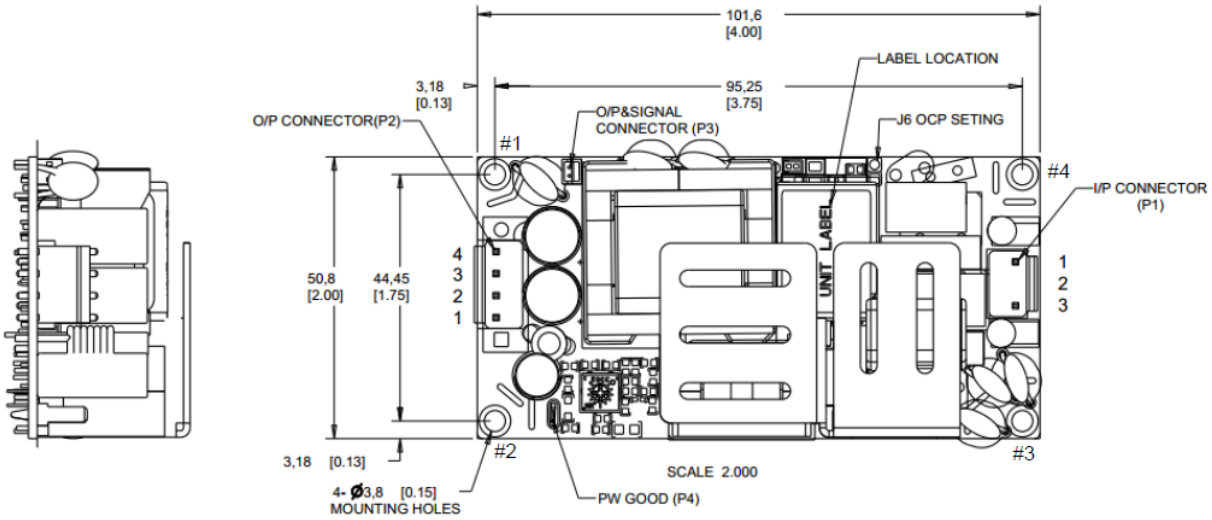


**Figure 54**

**Table 13-1** Connector Pin-Out for ASM201-240-BNH-PF1

| TERMINAL ASSIGNMENTS | | |
|---|---|---|
| P1 | 1 | ACL |
| | 2 | NC |
| | 3 | ACN |
| P2 | 1 | DC_OUTPUT+ |
| | 2 | DC_OUTPUT+ |
| | 3 | DC_OUTPUT- |
| | 4 | DC_OUTPUT- |
| P3 | 1 | 12V_AUX+ |
| | 2 | 12V_AUX- |
| P4 | 1 | PW GOOD |

## *13.2 Motor Details*

### 13.2.1 Azimuth

- Samsung DC31-00111A stator, DC31-00112A rotor
- 36 poles
- 3-phase
- Mean winding resistance = 5.9Ω

Applying 24VDC across two of the windings draws about 2A. The motor's holding force with this much power applied far exceeds what will be needed to drive the azimuth axis.

### 13.2.2 Elevation

- 42mm BLDC 24V (Nominal)
- Brushless 3 Phase DC Motor
- 8 poles

| | | 42BLS01 | 42BLS02 | 42BLS03 | 42BLS04 |
|---|---|---|---|---|---|
| Number of Poles | | 8 | | | |
| Number of Phase | | 3 | | | |
| Reted Voltage | VDC | 24 | | | |
| Reted Speed | | 4000 | | | |
| Continuous Stall Torque | N.m | 0.075 | 0.15 | 0.22 | 0.3 |
| Rated Torque | N.m | 0.0625 | 0.125 | 0.185 | 0.25 |
| Rated Power | W | 26 | 52.5 | 77.5 | 105 |
| Peak Torque | N.m | 0.19 | 0.38 | 0.56 | 0.75 |
| Peak Current | A | 5.4 | 10.6 | 15.5 | 20 |
| Line to Line Resistance | Ω | 1.8 | 0.8 | 0.55 | 0.28 |
| Line to Line Industance | mH | 2.6 | 1.2 | 0.8 | 0.54 |
| Torque Constant | Nm/A | 0.035 | 0.0355 | 0.036 | 0.0376 |
| Back E.M.F | Vrm/KRPM | 3.66 | 3.72 | 3.76 | 3.94 |
| Rotor Inertia | g.cm² | 24 | 48 | 72 | 96 |
| Body Length | | 41 | 61 | 81 | 101 |
| Weight | | 0.3 | 0.45 | 0.65 | 0.8 |

**Figure 55** Elevation motor parameters [17] for baseline motor (42BLS02) and higher-torque motor (42BLS03) if needed

The elevation axis motor drives a 80:1 harmonic drive. The main issue determining the minimum torque required for the elevation motor is the no-load starting torque required by the drive itself.

[17] U26749 TS-42BLS 3-Phase DC Motor.pdf.

# 14 Appendix: Mount Calibration Using Live-Sky Measurements

The material which follows is admittedly a bit ahead of my present needs, but also a topic which is of great interest to me.

Live-sky assessment using the night-time star field is a particularly easy (and accurate) way in which to assess and calibrate the telescope's performance. Given the precise time of day along with sighting on easily observable bright stars can provide a wealth of information in this regard. Later on once full axes motor control is operational, this same star field calibration method can and will be fully automated.

Star coordinates in the sky are given in terms of *declination* and *right-ascension* quantities. These are similar to earth-bound latitude and longitude, but are tied to the celestial sphere rather than the earth. It is fairly easy to convert between this coordinate system and an azimuth/elevation based system once a set of cardinal points have been identified.

At the beginning of this effort, telescope mount attitude (with respect to horizontal in two orthogonal directions) and even the precise degree of orthogonality between the azimuth and elevation axes will not be known. Add to this the as-yet unknown imprecision of the optical encoders, and a fair amount of effort can be required to quantify all of the system errors.

Navigating around the night time sky is remarkably complicated if precision on the order of a few arc-seconds[18] is desired. This is in part due to the large (new) vocabulary involved; terms like ecliptic, right-ascension, declination, precession, sidereal time, etc.

There are two primary motivations for using live-sky star navigation:

1. Exhaustive star catalogs are available which provide very accurate position information for thousands of stars, but the coordinate information is present in celestial coordinates
2. Knowing how to transform stellar celestial coordinates to local azimuth/elevation coordinates makes it possible to determine alignment errors and mechanical precision errors in the telescope mount.

## *14.1 Stellarium*

One way to obtain relatively precise stellar celestial coordinate information is to use a free public-domain computer program named Stellarium. Stellarium is remarkably easy to use. It can be used to find a stars right-ascension and declination coordinates in the sky for any time and date of interest, and its visual graphics are excellent.

[18] One arc-second is 1/3600 of $1^{o}$. One complete circle represents 360 x 3600 = 1,296,000 arc-seconds. In binary form, this represents about 20.31 bits or precision required.

## *14.2 Some Astronomy Terminology*

Quoting from [15]:

> **Right Ascension:** is measured (from 0 to 24 hours, sometimes from $0^o$ to $360^o$) from the vernal equinox, positive to the east, along the celestial equator. *A related quantity, sidereal time is defined as the right ascension of the stars that are on the meridian at a particular instant. Because of this definition, sidereal time can be determined directly by observation of the sky.*[19]
>
> **Declination:** is measured (from $0^o$ to $\pm 90^o$) from the equator, positive to the north, negative to the south.

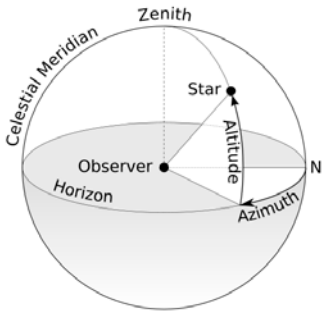


**Figure 56** Azimuth and altitude (elevation)[20]

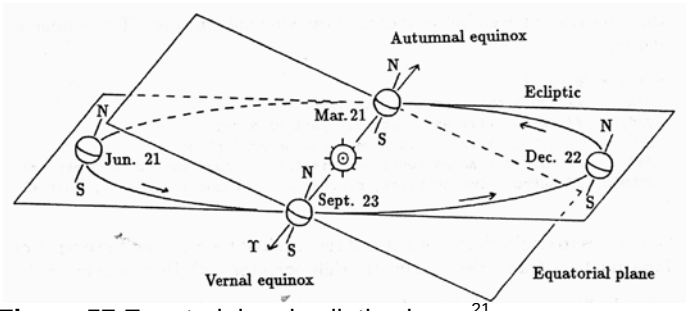


**Figure 57** Equatorial and ecliptic planes[21]

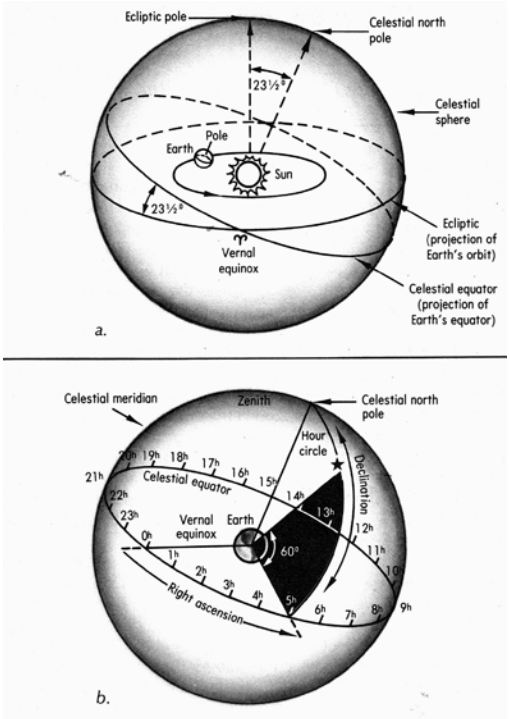


**Figure 58** (a) Celestial sphere and (b) right ascension and declination on the celestial sphere[22]

[19] [13], §3.3.
[20] Wikipedia.
[21] [13], Figure 2.2.

Equatorial coordinates (Right Ascension, Declination) can be converted to Azimuth/Elevation coordinates as[23]

**Table 2** Definition of Terms

| Quantity | Definition |
|---|---|
| A | Azimuth |
| E | Elevation |
| $\alpha$ | Right Ascension |
| $\delta$ | Declination |
| h | Hour Angle |
| $\varphi_o$ | User's Latitude |

$$A = \tan^{-1}(y, x) \tag{11}$$

with

$$\begin{aligned} x &= \cos(\varphi_o)\sin(\delta) - \sin(\varphi_o)\cos(\delta)\cos(h) \\ y &= \cos(\delta)\sin(h) \\ h &= local\_sidereal\_time - \alpha \end{aligned} \tag{12}$$

and

$$E = \sin^{-1}\left\{\sin(\varphi_o)\sin(\delta) + \cos(\varphi_o)\cos(\delta)\cos(h)\right\} \tag{13}$$

Given a star's right ascension and declination, the ideal azimuth and elevation angles can be computed using (11) and (13) for each star observed.

The observed Azimuth and Elevation $(A', E')$ for each star at the given times *h* are known from observation using the telescope mount with its associated alignment errors, and in general will be different than the true value $(A, E)$. A minimum of 3 separate star observations must be made in order to solve for the transformation matrix which relates $(A', E')$ to $(A, E)$ since 3 coordinate rotations are involved: yaw, pitch, and roll. The actual yaw, pitch, and roll angles do not need to be found, only the transformation matrix *R* as given by

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = [R] \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{14}$$

for true star positions $\vec{u} = (x, y, z)^T$ and observed star positions $\vec{v} = (x', y', z')^T$ where

$$[R] = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix} \tag{15}$$

[22] Figure 3.1 of [14].
[23] u28162_Coordinate_Transformation-Arcturus.mcd.

Assume the observed star positions are given by the $\overrightarrow{v_k}$ and the true (transformed) positions are given by $\overrightarrow{u_k}$ for $k \in \{1,2,...K\}$. Then (14) can be used to form

$$\left[\overrightarrow{v_1},\overrightarrow{v_2},...,\overrightarrow{v_K}\right] = R\left[\overrightarrow{u_1},\overrightarrow{u_2},...,\overrightarrow{u_K}\right] \tag{16}$$

This can be transformed to

$$\left[\overrightarrow{v_1},\overrightarrow{v_2},...,\overrightarrow{v_K}\right]\left[\overrightarrow{u_1},\overrightarrow{u_2},...,\overrightarrow{u_K}\right]^T = R\left[\overrightarrow{u_1},\overrightarrow{u_2},...,\overrightarrow{u_K}\right]\left[\overrightarrow{u_1},\overrightarrow{u_2},...,\overrightarrow{u_K}\right]^T \tag{17}$$

meaning the solution for *R* is given by[24]

$$\left[\overrightarrow{v_1},\overrightarrow{v_2},...,\overrightarrow{v_K}\right]\left[\overrightarrow{u_1},\overrightarrow{u_2},...,\overrightarrow{u_K}\right]^T \left\{\left[\overrightarrow{u_1},\overrightarrow{u_2},...,\overrightarrow{u_K}\right]\left[\overrightarrow{u_1},\overrightarrow{u_2},...,\overrightarrow{u_K}\right]^T\right\}^{-1} = R \tag{18}$$

$$R = R_{vu}R_{uu}^{-1} \tag{19}$$

In order to see how well the system is initially aligned, it is still useful to convert the transformation matrix *R* back to the constituent rotation elements as initially given by (15). At first glance, this may seem to be a difficult task to perform, but upon expanding (15) this becomes

$$R = \begin{Bmatrix} \cos(\alpha)\cos(\beta) & \begin{bmatrix}-\sin(\alpha)\cos(\gamma)+\\ \cos(\alpha)\sin(\beta)\sin(\gamma)\end{bmatrix} & \begin{bmatrix}\cos(\alpha)\sin(\beta)\cos(\gamma)+\\ \sin(\alpha)\sin(\gamma)\end{bmatrix} \\ \sin(\alpha)\cos(\beta) & \begin{bmatrix}\cos(\alpha)\cos(\gamma)+\\ \sin(\alpha)\sin(\beta)\sin(\gamma)\end{bmatrix} & \begin{bmatrix}\sin(\alpha)\sin(\beta)\cos(\gamma)-\\ \cos(\alpha)\sin(\gamma)\end{bmatrix} \\ -\sin(\beta) & \cos(\beta)\sin(\gamma) & \cos(\beta)\cos(\gamma) \end{Bmatrix} \tag{20}$$

Straight-forward inspection reveals

$$\begin{aligned} \beta &= -\sin^{-1}\left(R_{3,1}\right) \\ \gamma &= \tan^{-1}(y,x) = \tan^{-1}\left(R_{3,2},R_{3,3}\right) \\ \alpha &= \tan^{-1}\left(R_{2,1},R_{1,1}\right) \end{aligned} \tag{21}$$

Telescope mount alignment errors are relatively easy to model. Errors due to any non-orthogonality between the azimuth and elevation planes, elliptical rather than circular symmetry or higher-order irregularities are correspondingly more difficult to assess.

[24] u28090_coord_transformation.m, u28090_coord_transformation.m.

## *14.3 Digressing: Kelper's Problem*

Using the night time star field as a means for precision measurement and or calibration naturally involves using the precision coordinates of the stars within some frame of reference. The underlying theory is intimately tied to astrometry and navigation. These fields are complicated for primarily two reasons: (1) as technology has advanced, so has astrometry and terms having one meaning 50 years ago have sometimes had their meanings changed (e.g., use of the word epoch) and (2) a large vocabulary of potentially unfamiliar terms can get involved like Julian date, ecliptic, right-ascension and declination, mean longitude, mean anomaly, true anomaly, sidereal time, precession, osculating elements, etc. While I do not want to get overly bogged down in astrometry-related details, some engagement is unavoidable.

The[25] osculating orbit of an object in space at a given moment in time is the gravitational Kelper orbit (typically a conical shape like an ellipse) that it would have around a central body (e.g., star) if all perturbations were absent. An osculating orbit and the object's position upon it can be fully described by the six standard Kepler orbital elements (osculating elements), which are easy to calculate as long as one knows the object's position and velocity relative to the central body[26]. The orbital elements stem from the original work of Johannes Kelper in the 18th century and include six parameters:

1. Eccentricity
2. Semi-major axis
3. Inclination of ascending node
4. Longitude of the ascending node
5. Argument of periapsis
6. True anomaly

It is worthwhile to at least define these terms since a form of these parameters is used to this day in the GPS system to describe the satellite orbits in a compact ephemeris form. The GPS chips in our phones intimately rely upon this orbital modeling to simplify calculations while maintaining good navigational precision.

Again relying upon Wikipedia, two elements define the shape and size of the ellipse:

- Eccentricity ($e$)—shape of the ellipse, describing how much it is elongated compared to a circle (not marked in diagram).
- Semi-major axis ($a$) — the sum of the periapsis and apoapsis distances divided by two. For classic two-body orbits, the semi-major axis is the distance between the centers of the bodies, not the distance of the bodies from the center of mass. In the context of earth-orbiting satellites, the periapsis and apoapsis distances do not usually include the radius of the earth so some caution is needed.

Two elements define the orientation of the orbital plane in which the ellipse is embedded:

- Inclination ($i$) — vertical tilt of the ellipse with respect to the reference plane, measured at the ascending node (where the orbit passes upward through the reference plane, the green angle $i$ in the diagram). Tilt angle is measured perpendicular to line of intersection between orbital plane and reference plane. Any three points on an ellipse will define the ellipse orbital plane. The plane and the ellipse are both two-dimensional objects defined in three-dimensional space.
- Longitude of the ascending node ($\Omega$) — horizontally orients the ascending node of the ellipse (where the orbit passes upward through the reference plane, symbolized by ☊) with respect to the reference frame's vernal point (symbolized by ♈). This is measured in the reference plane, and is shown as the green angle $\Omega$ in the diagram.

[25] https://en.wikipedia.org/wiki/Orbital_elements
[26] https://en.wikipedia.org/wiki/Osculating_orbit

The remaining two elements are as follows:

- Argument of periapsis ($\omega$) defines the orientation of the ellipse in the orbital plane, as an angle measured from the ascending node to the periapsis (the closest point the satellite object comes to the primary object around which it orbits, the blue angle $\omega$ in the diagram).
- True anomaly ($\nu$, $\theta$, or $f$) at epoch ($t_0$) defines the position of the orbiting body along the ellipse at a specific time (the "epoch").

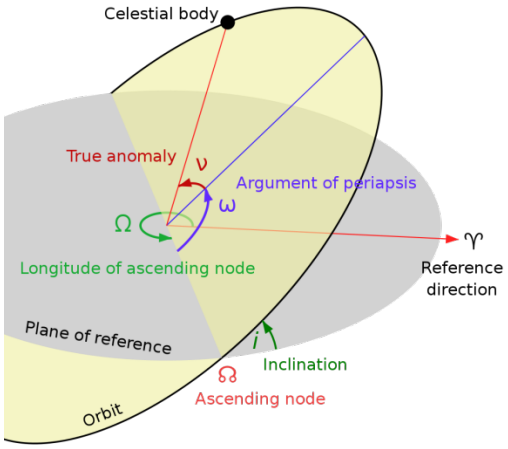


**Figure 59** Kepler's orbital parameters

It is generally not possible to pick up a book about orbital mechanics without finding a solution to Kepler's equation for elliptical orbits [13], [15], [16], [17], and [18]. Although the elliptical orbit shape can be defined with a closed-form solution, the orbital time behavior cannot as it requires a transcendental equation to be repeatedly solved.

Having some understanding of Kepler's 3-laws of planetary motion is key to understanding the mathematical details which follow. Kepler lived in the 18th century before even a notion of today's computers existed, and yet he confirmed his theories using intricate while exhausting hand calculations regarding the orbit of Mars based upon precision measurement data he obtained from Tycho Brahe. (Tycho's measurement precision was remarkably on the order of 1/30th the diameter of the moon using only naked-eye observations!)

**Kepler's 3-Laws**

- Planetary orbits about the sun are ellipses with the sun positioned at one of the two foci.
- A line drawn from the center of the sun to a given planet sweeps out equal areas in equal amounts of time.
- The orbital period for a planet is proportional to $a^{1.5}$ where *a* is the semi-major axis length of the orbit.

The two-dimensional polar form for an ellipse is given by

$$r = \frac{p}{1 + e\cos(\upsilon)} \tag{22}$$

where *e* is the orbit's eccentricity $0 \le e < 1$ and $\upsilon$ is the true anomaly as shown in Figure 60. It can be shown that parameter *p* is given by

 

$$p = a\left(1-e^2\right) \tag{23}$$

where the length of the semi-major axis is 2*a*. The orbital period is directly related to the area of the ellipse and is given by

$$T_p = \frac{2\pi}{\sqrt{GM_e}} a^{3/2} \sec \tag{24}$$

where *G* is the universal gravity constant (6.673e-11 in MKS units) and the mass of the earth is $M_e$ = 5.9802e24 kg.

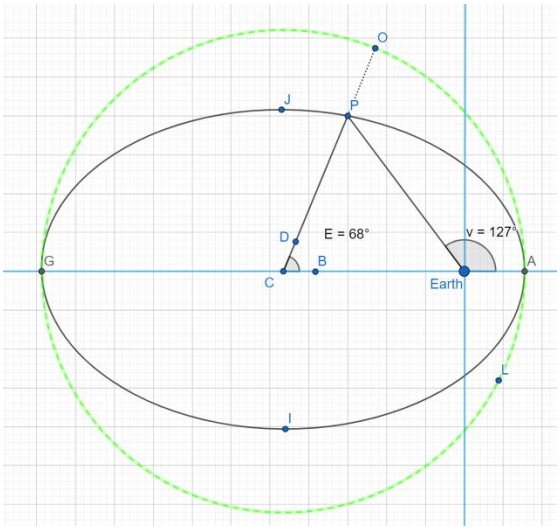


**Figure 60** Kepler's ellipse construction showing an ellipse and auxiliary circle (after [15]). Semi-major and semi-minor axes are given by segments $\overline{AG}$ and $\overline{IJ}$ respectively. The *true anomaly* angle is given by $\upsilon$ whereas the *mean anomaly* angle is given by *E*.

The key to solving the planet's orbital position versus time is to solve the *equation of Kepler* which is given by

$$E = \left(\frac{2\pi}{T_p}\right)\left(t - t_o\right) + e\sin\left(E\right) \tag{25}$$

in which the starting time $t_o$ can be taken to be zero seconds without loss of generality. It can be shown that the relationship between *E* and $\upsilon$ is given by

$$\tan\left(\frac{\upsilon}{2}\right) = \sqrt{\frac{1+e}{1-e}} \tan\left(\frac{E}{2}\right) \tag{26}$$

The solution to (25) for each time-step $t_k$ is frequently solved using an iterative Newton-Raphson approach. An error function *f*( ) is first defined using (25) as

$$f\left(E\right) = E - \frac{2\pi}{T_p} t_k - e\sin\left(E\right) \tag{27}$$

from which the first derivative follows as

$$\frac{df}{dE} = 1 - e\cos\left(E\right) \tag{28}$$

Starting with an initial estimate for *E* and $t_k$ (convenient to use either the apogee or perigee position), and improved estimate for *E* follows as

$$\begin{aligned} E' &= E - \frac{f\left(E\right)}{df\ /\ dE} = E - \frac{E - n_p t_k - e\sin\left(E\right)}{1 - e\cos\left(E\right)} \\ &= \frac{n_p t_k - e\left[E\cos\left(E\right) - \sin\left(E\right)\right]}{1 - e\cos\left(E\right)} \end{aligned} \tag{29}$$

where $n_p = 2\pi / T_p$ for convenience. Equation (29) can be iterated as many times as desired for better precision. If a complete orbit is to be computed, using the previous value of *E* at time $t_{k-1}$ serves as an excellent initial estimate for $E_{k+1}$.

**Side Note:**

> The simple form of Kepler's equation (25) immediately suggests an alternative computational approach may be used. I have not seen this approach suggested in the literature (probably because of its poorer precision than the method just described), but if care is exercised with the numerical integration which must follow, this approach will yield good results as well. Implicitly differentiating (25) with respect to time results in
>
> $$\frac{dE}{dt} = \frac{2\pi}{T_p}\frac{1}{1 - e\cos\left(E\right)} \tag{30}$$
>
> This result can be numerically integrated using Backward-Euler or a higher order implicit integration method (e.g., Tustin's method). The most simple Forward-Euler integration method is discouraged because of its poor accuracy. The precision-purist could even consider using a 4th-order Runge-Kutta method as well.

An example is helpful in putting the gravitational forces involved into context.

Newton's Law of Gravitation makes it straight forward to calculate the force exerted by an object situated in a solar orbit. For an example, consider a 200 pound (mass) man who is out in space in the earth's orbit at 93 million miles away from the sun. The gravitational attraction force exerted by the sun is given by Newton's equation where the force directed toward the sun is

$$F = G_o \frac{M_{sun} m_{man}}{r^2} \tag{31}$$

The parameter values in MKS units are given in Table 14-3. The result attractive force is only 0.1207 pounds, or equivalently 1/1657 that of the earth's pull on the same 200 pound (mass) man.

**Table 14-3** Gravitational Attraction Parameters (MKS units)

| | | |
|---|---|---|
| $G_0$ | 6.674e-11 | gravitational constant |
| $M_{sun}$ | 1.98847e30 | solar mass, kg |
| $m_{man}$ | 90.718 | mass of 200 lb man, kg |
| $r$ | 1.4967e11 | radius of orbit, m |
| $F$ | Calculated | attraction force of sun on the man, directed toward the sun's center, Newtons |

# 15 Next Steps

With the mechanical hardware and electronics now all but completed, the next steps are primarily involved with software development. While I plan to utilize a lot of the code (C# and DSP C) I have already developed, I need to do a fair amount of re-architecting to accommodate (i) operating both azimuth and elevation encoders, (ii) code mapping changes driven by now hosting two DRV8301 Booster Packs on the TMS320F28379D Launchpad board rather than one, and (iii) developing a friendly GUI interface and supporting calculations to do Live Sky mount calibrations.

I will also be making full use of the TMS320F28379D's built-in CLA hardware accelerators in order to off-load the main processors while also reducing the computational delay associated with each encoder sample. Ideally, new angular measurements will be made at a 12.5 kHz rate so that control loop bandwidths on the order of several hundred Hertz will be readily achievable. This bandwidth may seem excessive, but holding pointing accuracy to a few arc-seconds with even slight air (wind) movement around the telescope is not a trivial matter.

One of the major new areas ahead of me is the 3-phase precision motor control needed for the mount. It is easy to get a bit overwhelmed with the enhanced pulse width modulator (ePWM) programming of the TMS320F28379D as there are 7,398 lines of code in the epwm.h file, and 1,650 lines of code in the hrpwm.h file. Example files focusing on the ePWM blocks are fairly minimal as well, and one must dive into the full-control code examples (e.g., IDDK_PM_Servo_F28379x) to see the full scope of the ePWM blocks in use.

The technical reference manual [10] devotes 89 pages to the ePWM function plus an additional 25 pages to discuss the high-resolution pulse width modulator function.The register map discussion is 132 pages long itself.

The F28379D's ePWM modules provide far more features than required for my motor control application. This should lead to a lot of programming simplification.