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

Part 6A: Completing the Hardware

James A Crawford

Synopsis

A lot has happened since my last project-update. 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.
- *C# GUI and TI software were modified to accommodate the two 3-phase DRV8301 motor drivers and two optical encoders. This required a second DRV8301* Booster Pack to be hosted on the TMS320F28379D DSP in order 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.

- *Software additions were made to support live azimuth/elevation data collection on sighted objects.*
- *Software additions were also made to support the first installment of full 3-phase azimuth/elevation motor control.*
- *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.*
- *Bulleted items in light blue will be reported on in Part 6B.*

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.

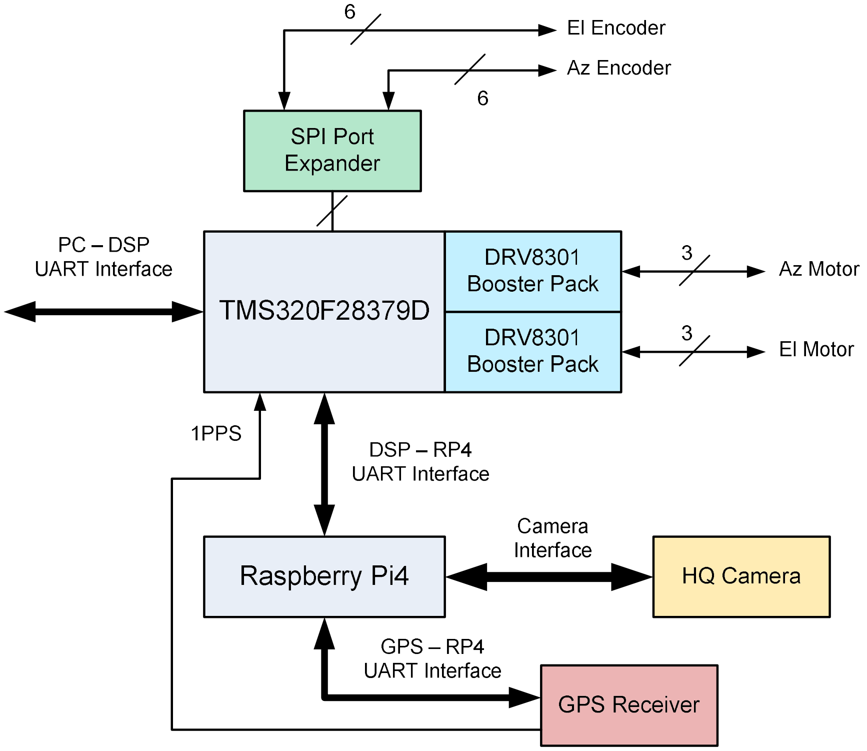


Figure 1 Multi-processor approach going forward to accommodate (i) Az & El 3-phase motor control, (ii) Az & El optical encoders, and (iii) Raspberry Pi4 to host an HQ image sensor and GPS receiver[2]

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 [2], 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 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.

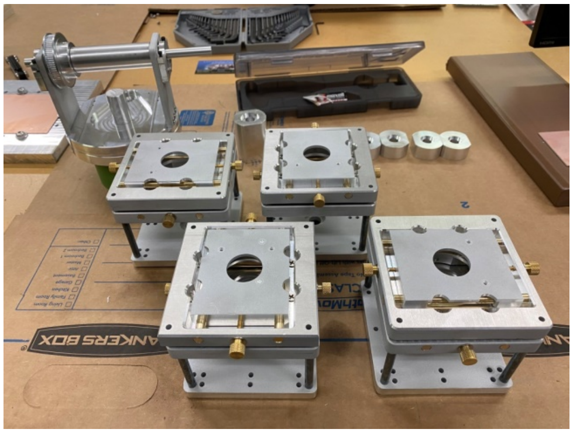


Figure 2 Replicated optical encoder mounts to support two additional telescope efforts planned for the future

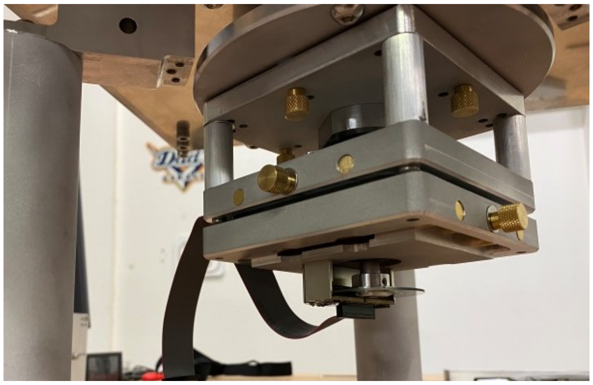


Figure 5 Azimuth encoder installed below the azimuth axis motor

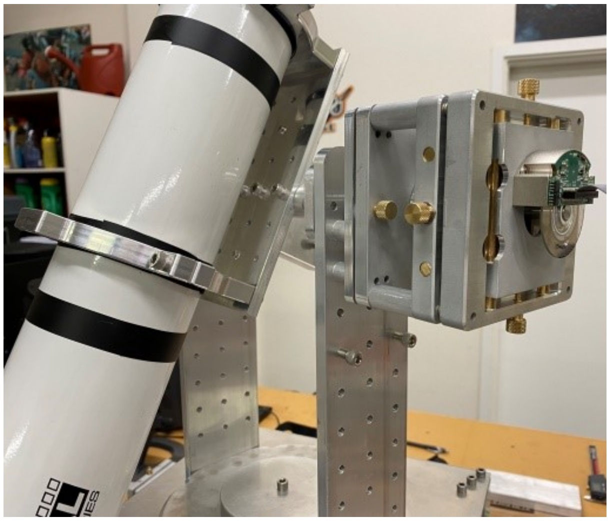


Figure 6 Elevation encoder installed on one side of the axis

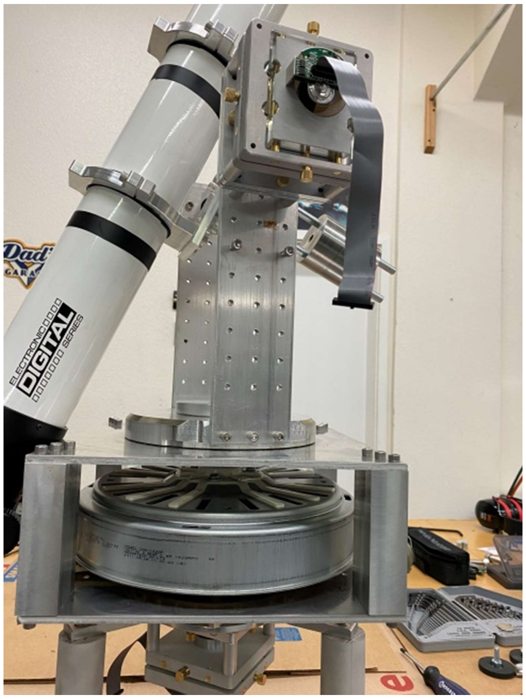


Figure 3 Optical encoders just mounted to the telescope mount

Figure 4 In the workshop

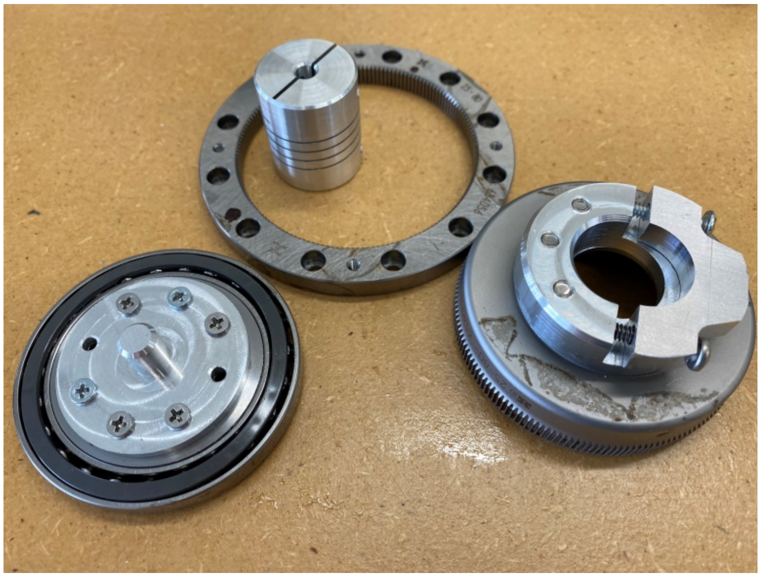


Figure 7 Primary harmonic drive components purchased on Ebay, further augmented with an axle clamp.

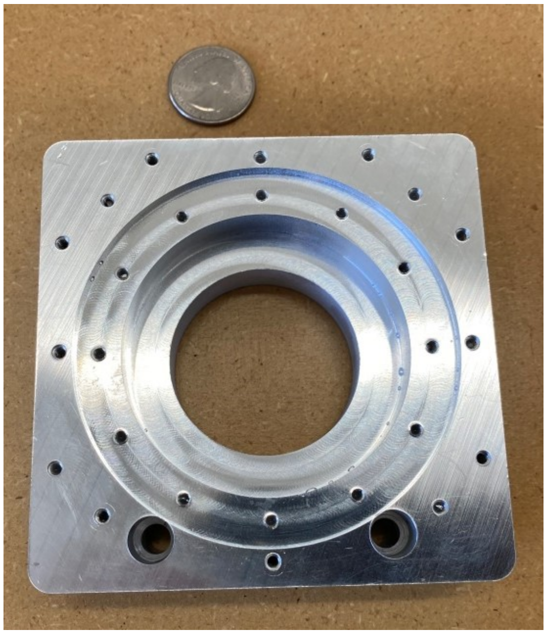


Figure 8 Part of the fabricated harmonic drive housing used to secure the components into their proper place

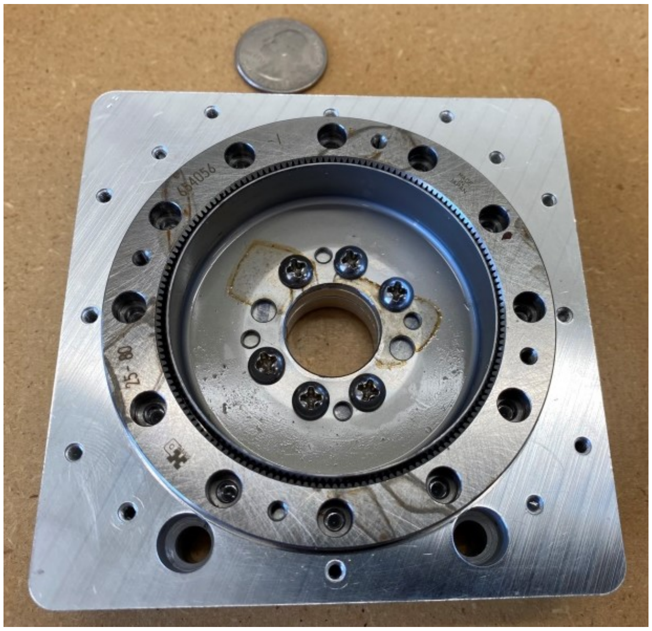


Figure 9 Two of the three major harmonic drive components temporarily installed into the housing of Figure 8

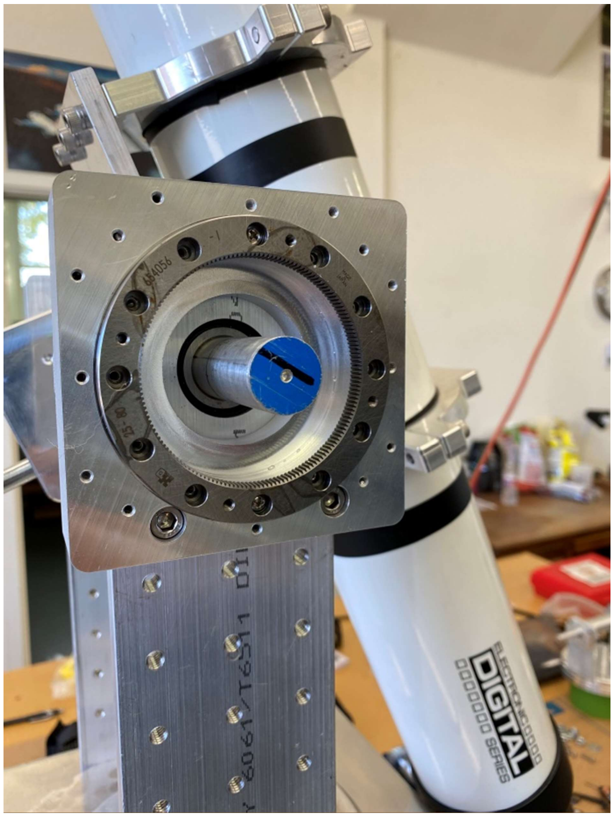


Figure 10 Harmonic drive assembly into the telescope mount

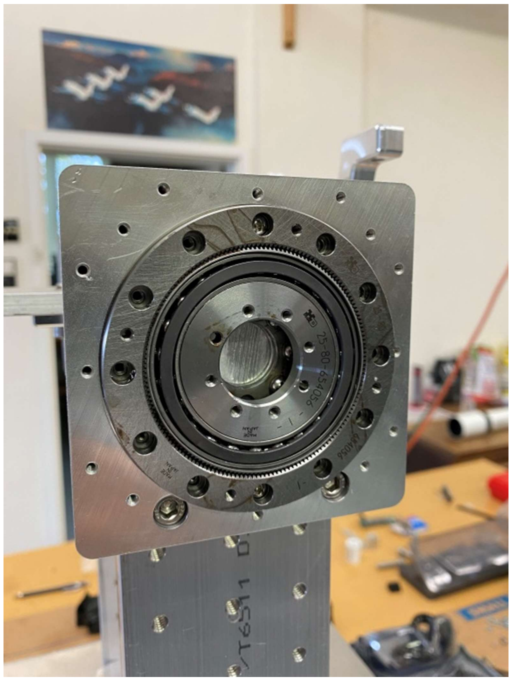


Figure 11 Interim assembly step

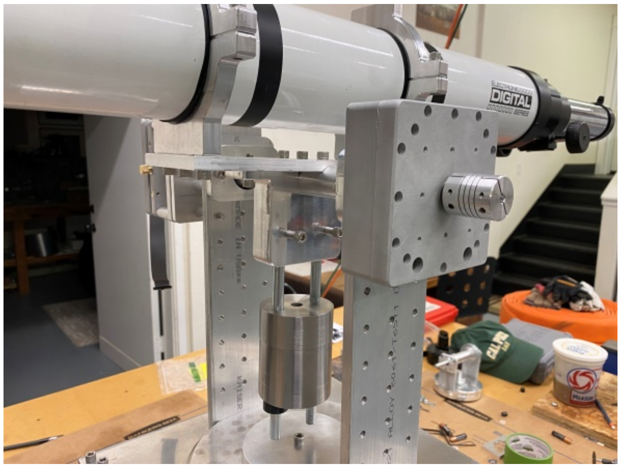


Figure 12 Side view of the encapsulated harmonic drive before installing the motor

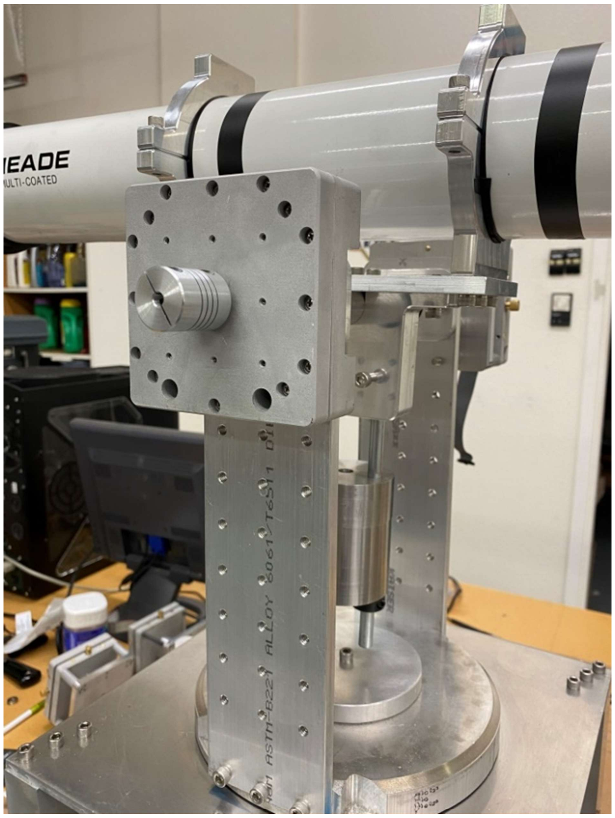


Figure 13 Slightly different view of Figure 12

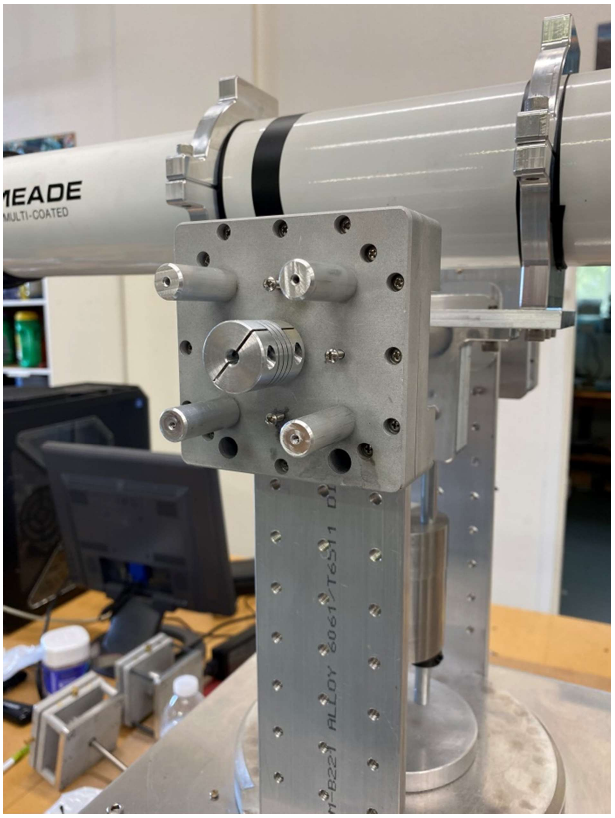


Figure 14 Harmonic drive with flex-coupling and motor offset posts

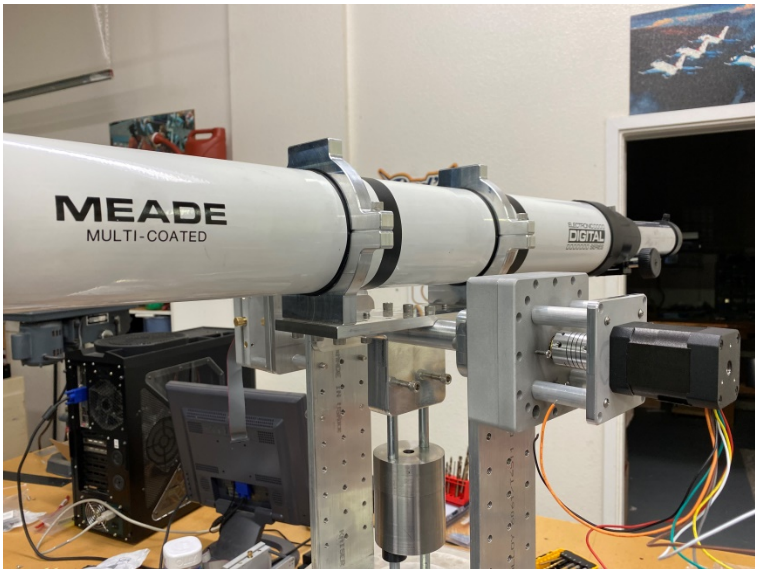


Figure 15 Fully assembled elevation harmonic drive and associated 3-phase motor

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

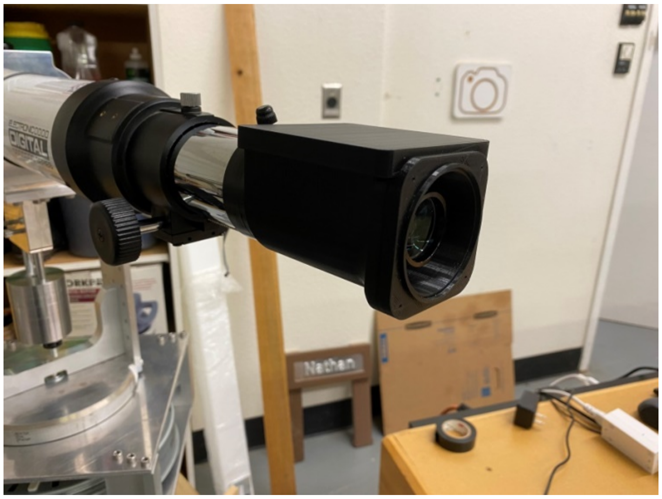


Figure 16 Focus tube and eyepiece mount adapter portion

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

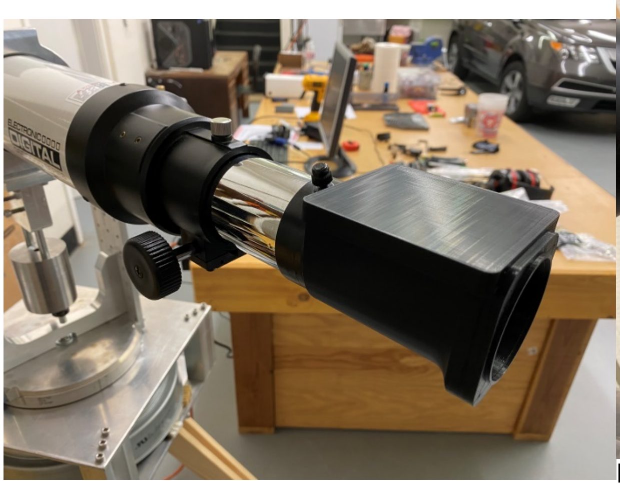


Figure 17 Side-view of Figure 16

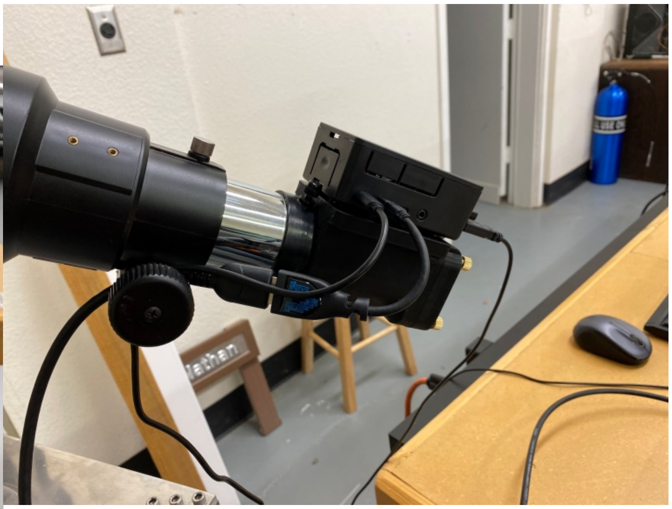


Figure 18 Fully assembled camera with HQ sensor portion of the adapter now attached, along with a Raspberry Pi4 atop to capture the sensor data. The GPS receiver will be piggy-backed atop the Raspberry Pi processor.

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.

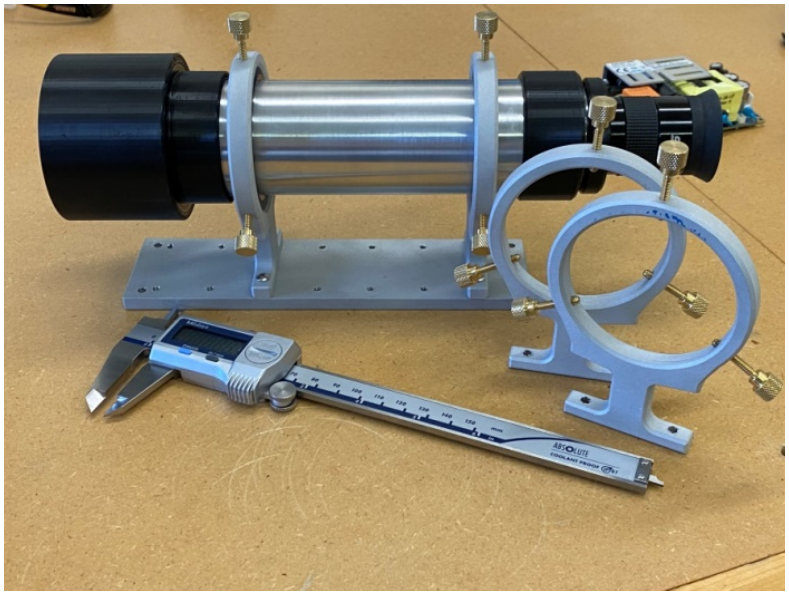


Figure 19 Finder telescope along with mounting rings and plate adapter

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.

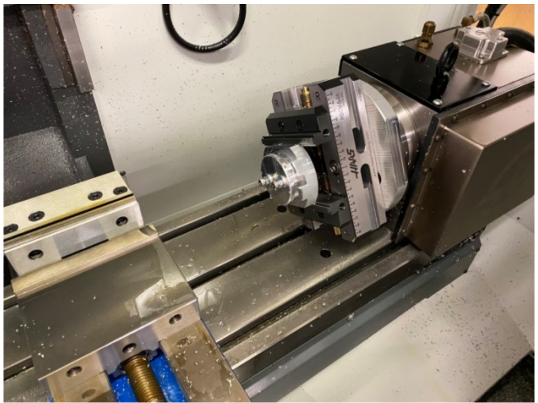


Figure 20 Fabrication jig situated on the mill's 4th axis

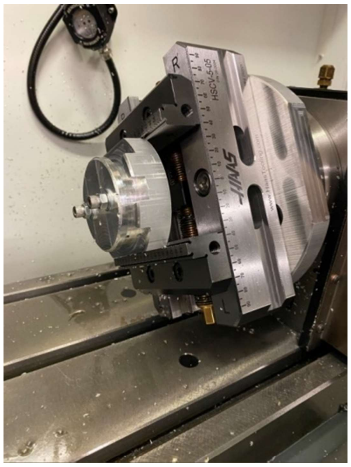


Figure 21 Close-up of Figure 20

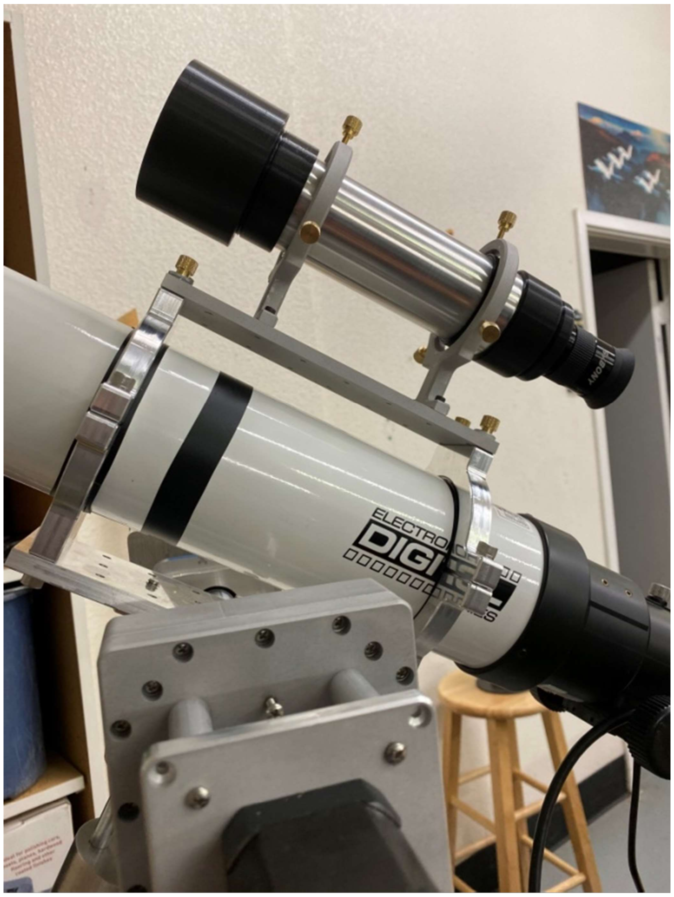


Figure 22 Finder telescope, ring mounts, and plate adapter installed on the main telescope

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.

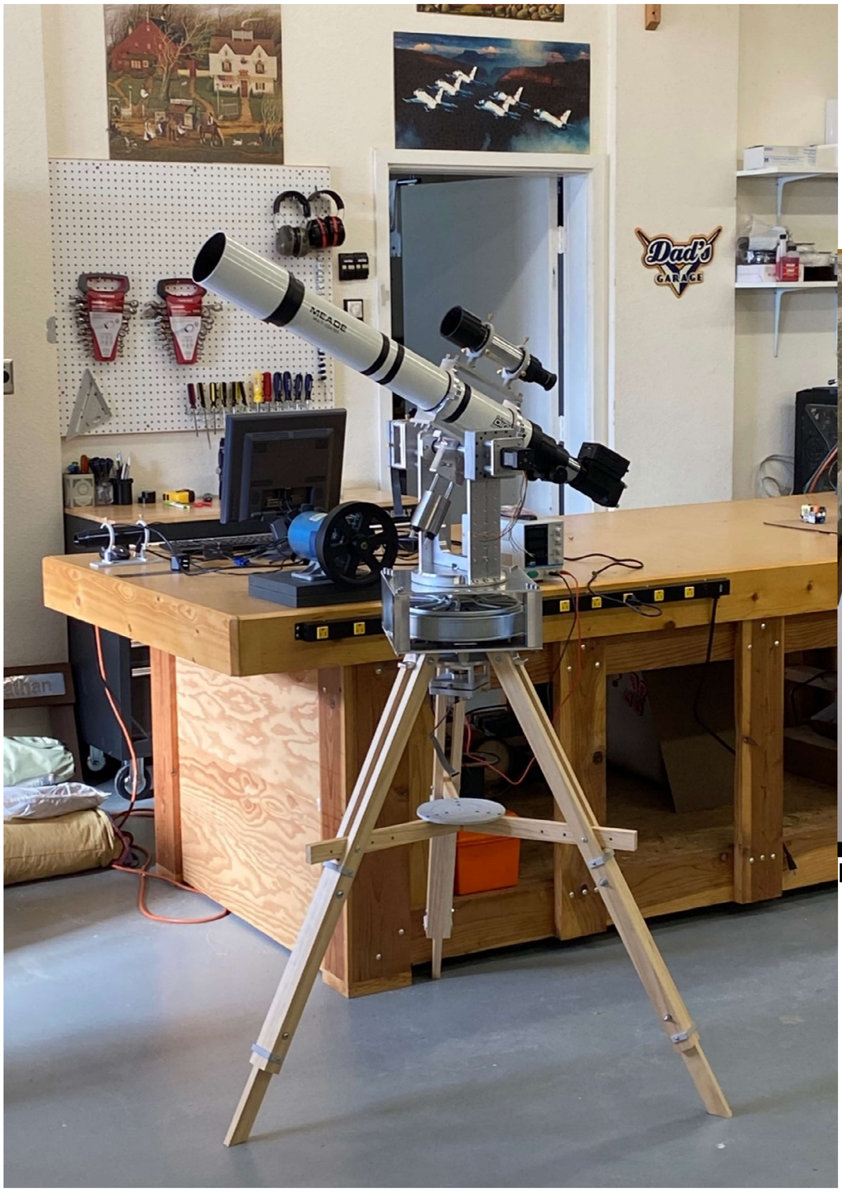


Figure 23 Fabricated telescope tripod with the full assembly. A 2 horsepower motor intended for a future telescope mirror grinding machine is shown on the benchtop.

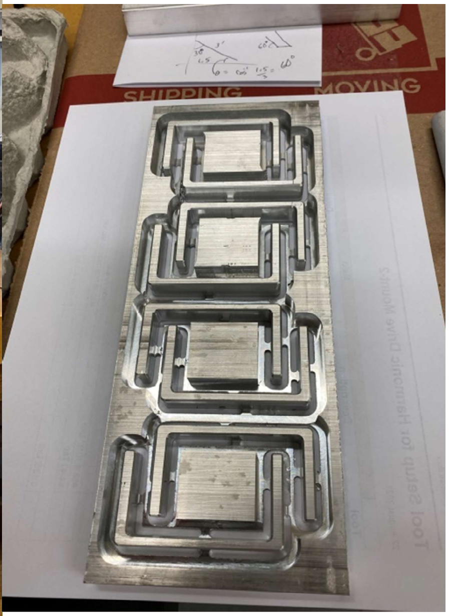


Figure 24 Milled leg-clamps

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 [3]. 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.

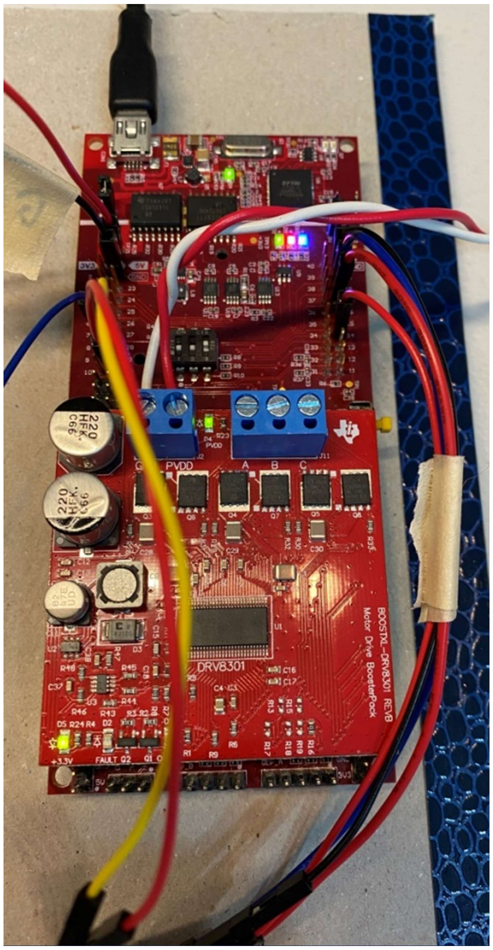


Figure 25 One DRV8301 Booster Pack installed atop the TMS320F28379D Launchpad XL DSP board. The twisted red-white pair bring in +12V whereas other voltages (3.3V and 5V) are created internally. In the final configuration, the red-white pair will provide 24 VDC to the circuitry.

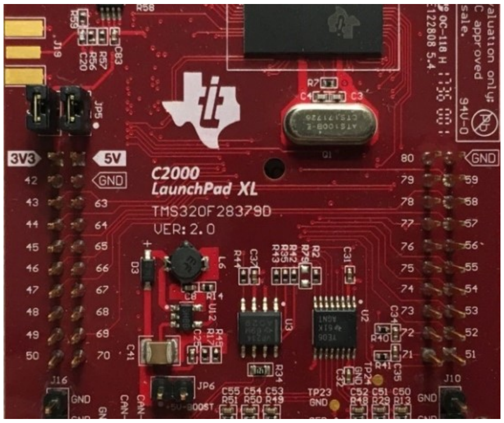


Figure 26 F28379D Launchpad PWB area furthest from the USB connector, showing from left to right, J5, J7, J8, and J6

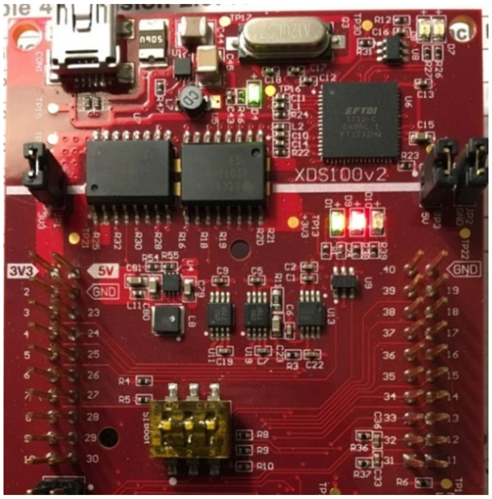


Figure 27 F28379D Launchpad PWB area closest to the USB connector, showing from left to right, J1, J3, J4, and J2

[4] SPI operation detailed at length in Chapter 18 of [7].

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

Table 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 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.

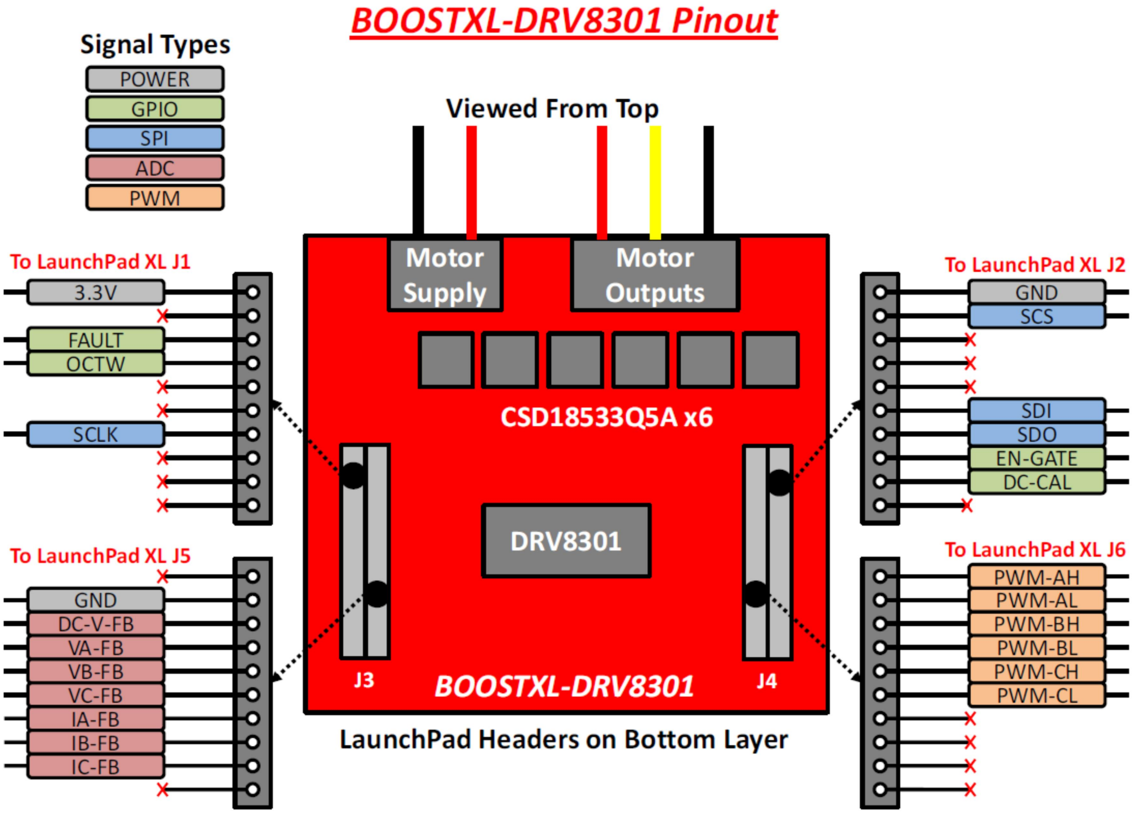


Figure 28 DRV8301 pin-out [5]

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 2 and Table 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 [5]:

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.

>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.

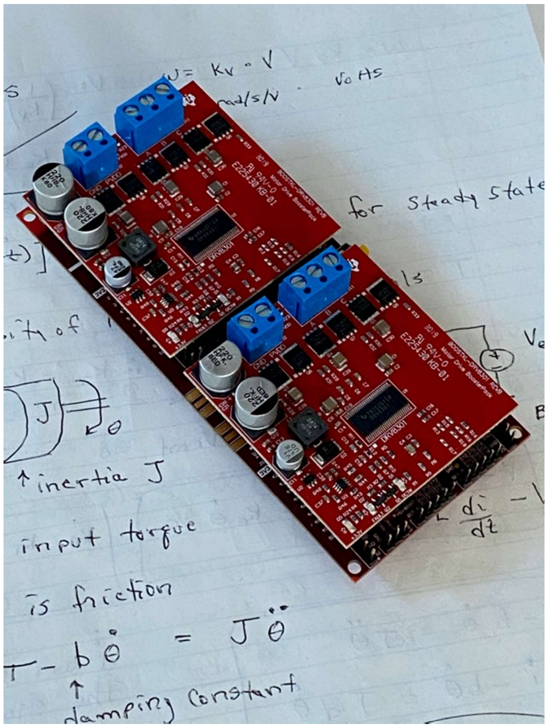


Figure 29 TMS320F28379D hosting two DRV8301 Booster Packs

Table 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	
FAULT	3				19	GPIO19 / SCIRXDB / SPISTEA*	
OCTW	4				18	GPIO18 / SCITXDB / SPICLKA	
x	5				67	GPIO67	
x	6				111	GPIO111	
SCLK	7				60	GPIO60 / SPICLKA / SPISIMOB	
x	8				22	GPIO22 / SCITXDB / SPICLKB	
x	9				105	GPIO105 / SCLA / SCIRXDD	
x	10				104	GPIO104 / SDAA / SCITXDD	
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	
x			33		16	OUTPUTXBAR7 / GPIO16 / SPISIMOA	
x			32			DAC1	
x			31			DAC2	
GND				20		GND	
SCS				19	61	GPIO61 / SPISIMOB / SPISTEA*	
x				18	123	GPIO123 / SD1_C1 / SPISOMIC	
x				17	122	GPIO122 / SD1_D1 / SPISIMOC	
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	

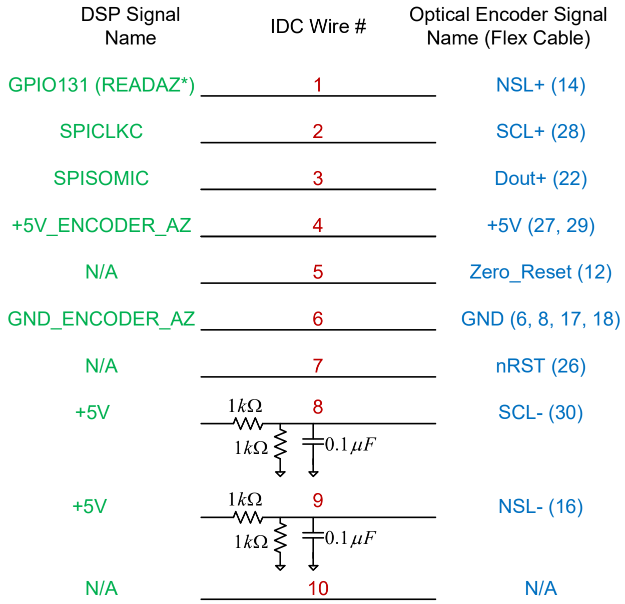
[5] From U26355 LAUNCHXL-F28379D Overview, SPRUI77C, August 2016, March 2019. Not all GPIO etc pins are mapped to J-connectors. See U27334 F28379D Datasheet sprs880k for details.

Table 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	
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	GPIO24 as SPISTE* for El Encoder
x	50				40	GPIO40 / SDAB	GPIO16 as SPISTE* for Az Encoder
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	166	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	

*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.



DSP Signal Name	IDC Wire #	Optical Encoder Signal Name (Flex Cable)
GPIO131 (READAZ*)	1	NSL+ (14)
SPICLKC	2	SCL+ (28)
SPISOMIC	3	Dout+ (22)
+5V_ENCODER_AZ	4	+5V (27, 29)
N/A	5	Zero_Reset (12)
GND_ENCODER_AZ	6	GND (6, 8, 17, 18)
N/A	7	nRST (26)
+5V	8	SCL- (30)
+5V	9	NSL- (16)
N/A	10	N/A

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*.

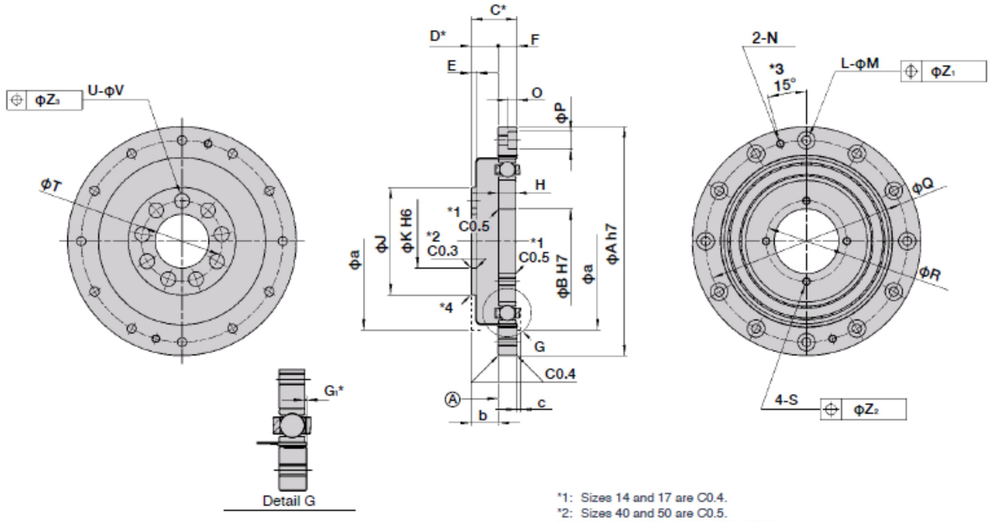
To be continued in Part 6B.

[6] Described in §18.2.1 of [7].
[7] From U27771_Part6_Figures.vsd.

7 References

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 IV: Optical Encoder and Elevation Drive,” U24933, August 2020.
3. ____________, “Photogrammetry for Non-Invasive Terrestrial Position/Velocity Measurement of High-Flying Aircraft- Part I,” 2017, U24866.
4.
5. Texas Instruments, “BOOSTAXL-DRV8301 Hardware User’s Guide,” SLVU974, Oct. 2013, U27331.
6. ________________, “The TMS320F2837xD Architecture: Achieving a new Level of High Performance,” SPRT720 Feb 2016, U27333.
7. ________________, “TMS320F2837xD Dual-Core Delfino Microcontrollers, Technical Reference Manual,” Jan. 2019, SPRUHM8H, U26369.
8. Elliott D. Kaplan and Christopher J. Hegarty, *Understanding GPS, Principles and Applications*, 2nd ed., Artech House, 2006.
9. Harmonic Drive LLC, “HarmonicDrive® Reducer Catalong”, U27619.

8 Appendix: Harmonic Drive Assembly



*1: Sizes 14 and 17 are C0.4.
*2: Sizes 40 and 50 are C0.5.
*3: Size 14 is 30° and Size 17 is 22.5°.
*4: Sizes 14 to 32 are C0.3 and 40 and 50 are C0.5.
* Refer to confirmation drawing provided by Harmonic Drive LLC.

Figure 31 Harmonic drive dimensions from [9]

Table 4 Dimension Excerpt from [9]

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

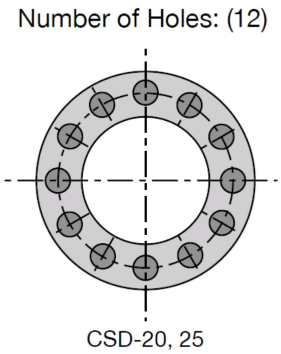


Figure 32 Mounting holes

9 Appendix: Azimuth Drive- Bottom Plate for Encoder Mount

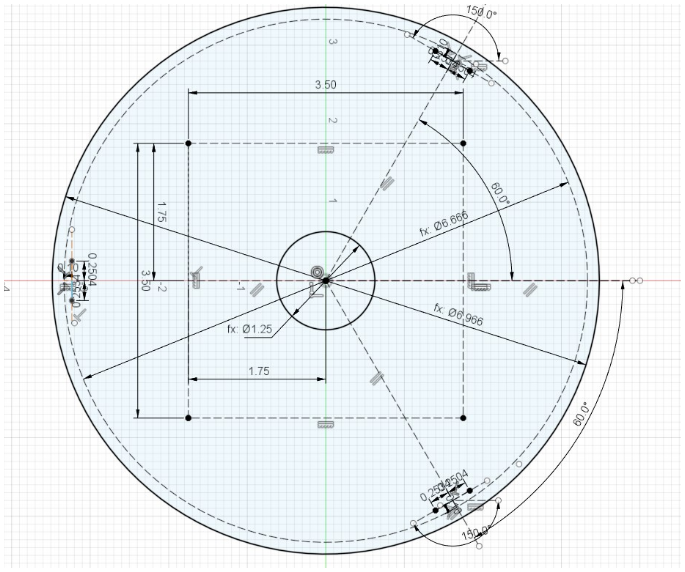


Figure 33 Telescope Az Encoder Mounting Plate

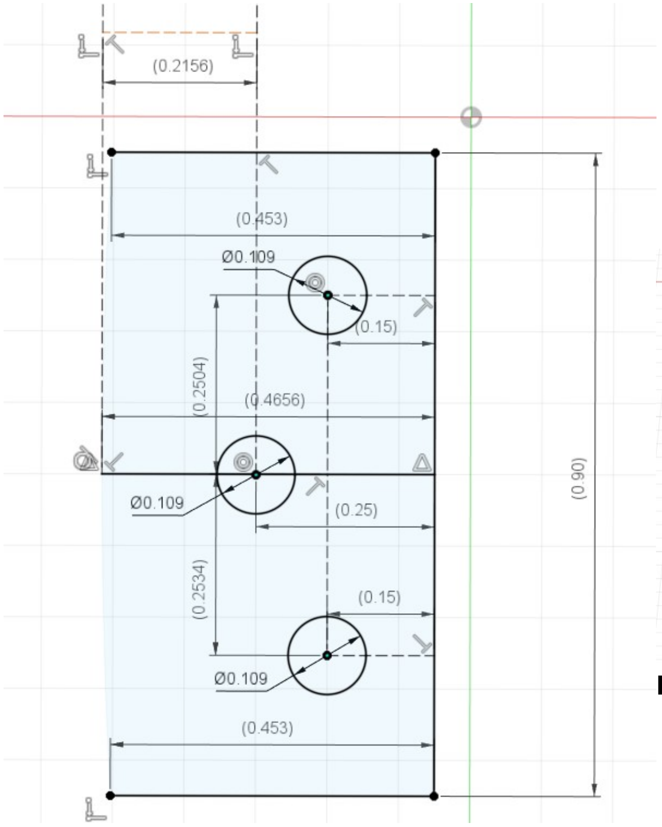


Figure 34 Detail for mounting hole region of “feet”

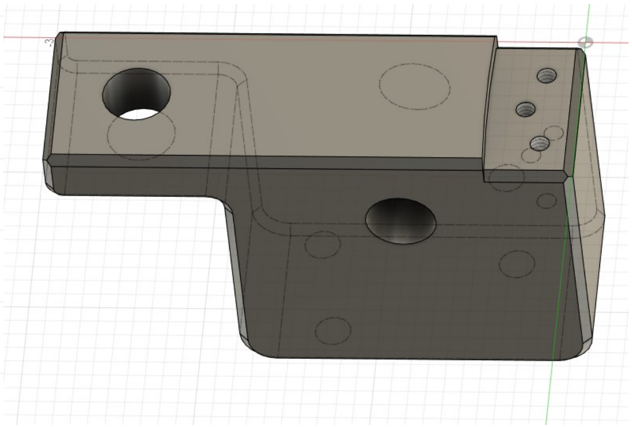


Figure 35 One of the “feet”

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.

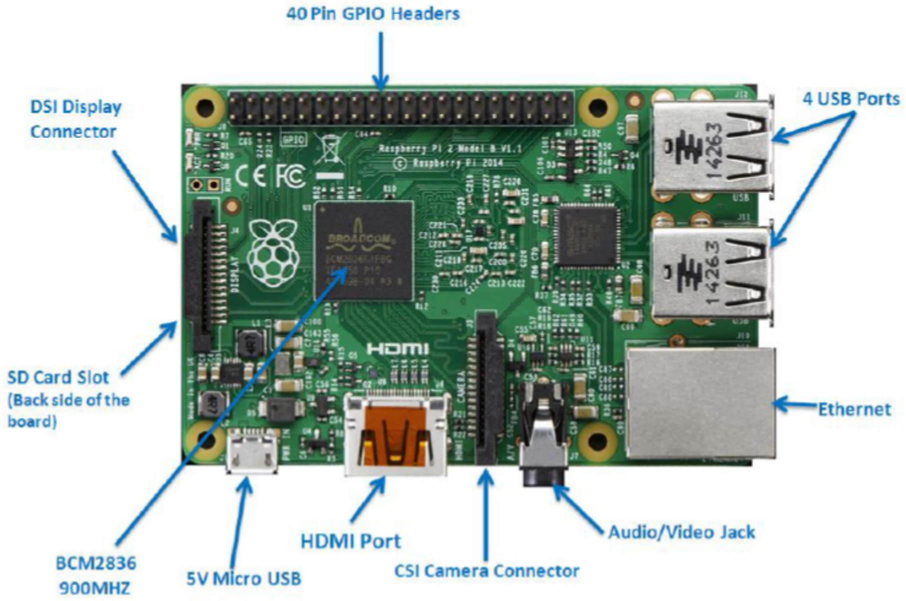


Figure 36 Raspberry Pi 4[8]

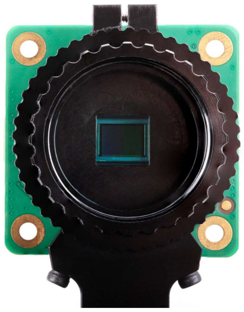


Figure 37 Front view of the HQ sensor[9]

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

[8] From U27684, Raspberry Pi Tutorial.

[9] From “HQ Camera Product Brief,”U27679.

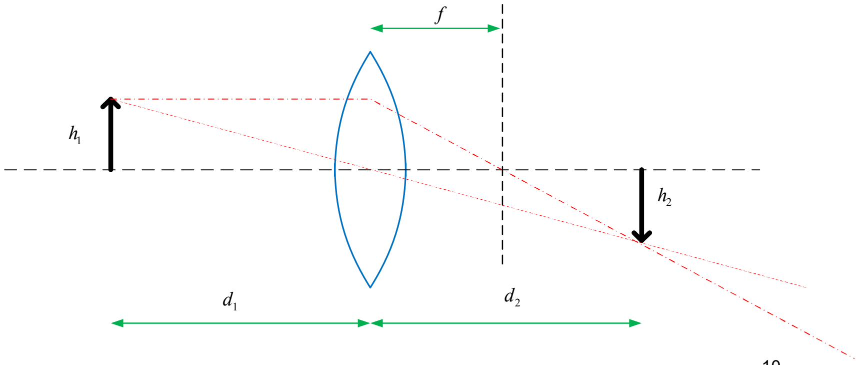


Figure 38 Simple arrangement using objective lens and CCD sensor at the focus[10]

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

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

$$d_2=\frac{fd_1}{d_1-f} \tag{3}$$

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 θ_1 is in radians. Since $d_1>>f$, the image height is simply given by

[10] U27712 CCD Telescope Optics.vsd.

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

Several values are computed in Table 5, illustrating how the CCD sensor size is more than filled for focal lengths greater than roughly 12”.

Table 5 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.

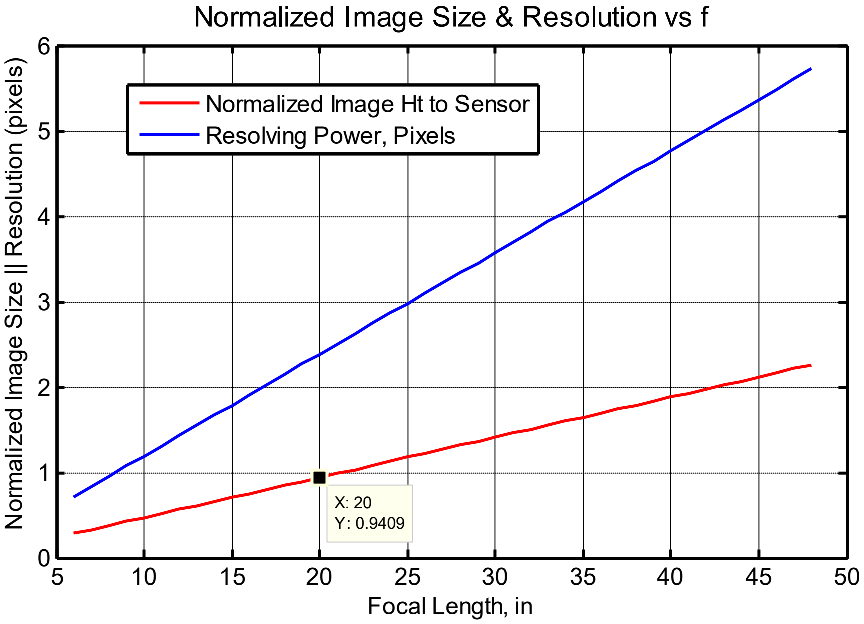


Figure 39 Resolving power and (moon) image size versus focal length[11]

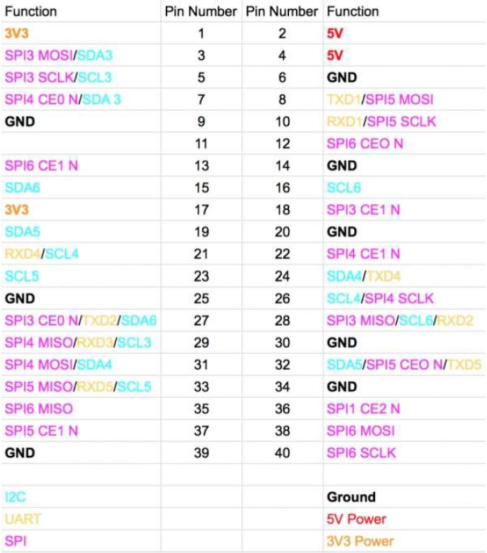
$$f_{eff} = \frac{f_1 f_2}{f_1 + f_2 - d} \tag{9}$$

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

[11] u27713_ccd_optics.m.

11 Appendix: Raspberry Pi4 External Interface

Table 6



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