30/05/23

Time: 1h

Tuesday, May 30, 2023 12:42 PM

Meeting w/ Nick

Focus : - Review previous work
- Search for computer
- Search for board level Camera

31/05/23

Time: 8h

Wednesday, May 31, 2023 7:58 AM

Review TRR report 8:00AM - 10

-

CCP-SP1-TRR-0160...

Things to research

- Cassegrain reflector ✓
- Fresnel lens ✓
- edge spread function ESF
- modulation transfer function - Quick MTF software
- Total Mass Loss TML
- pareto analysis
- UART * ✓
- UHF
- S-Band
- b-dot algorithm ✓
- magnetorquers ✓
- TLE File ✓

Research

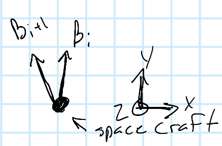
Magnetorquers

- Coil that generates a magnetic field
- Generated field creates a magnetic moment with earths magnetic field
- Magnetic moment measures an objects tendency to align with a magnetic field
- https://byjus.com/physics/magnetic-moment/#:~:text=Asked%20Questions%20%E2%80%93%20FAQs-,What%20Is%20Magnetic%20Moment%3F,moment%20is%20a%20vector%20quantity.
- So the magnetorquer generates a field that will generate a torque to align with earths magnetic field
- The toque is the cross product of the magnetic moment and earths magnetic field.

B-dot algorithm

Algorithm specifically for using magnetorquers to control a satellite

- Changes control signal to the magnetorquer to control alignment of the spacecraft
- $M_i = -k_i \dot{B}_i$
- Here, M is the magnetic moment of the magnetorquer, k is presumably some gain, and B dot is the derivative of earths magnetic field.

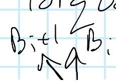


$$B_i = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\dot{B} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad M_i = -\dot{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

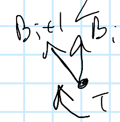
$$T = M \times B = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix} \quad 0\hat{i} - 0\hat{k} + 1\hat{j}$$

+ Torque in Z



Checks out

+ Torque in Z



Checks out

- The algorithm works by measuring earths magnetic field, comparing this to the previous time step to determine the rate of change, and defining the magnetorquer moment based off this.
- This cycle repeats to maintain alignment of the magnetorquers with earths magnetic field
- http://www.ece3sat.com/blog/2018-01-25-how-to-slow-down-rotation-the-bdot-algorithm/#:~:text=How%20does%20the%20B%2Ddot,law%20equivalent%20for%20rotational%20motions).

Does this keep the same face of the satellite facing earth or remove rotation from an inertial frame in space?

As the satellite is rotating around earth, from its reference, earths magnetic field is rotating. Since the B-dot algorithm aligns with earth, the satellite would also rotate. So it would been the same side facing down all the time

TLE File

https://en.wikipedia.org/wiki/Two-line_element_set

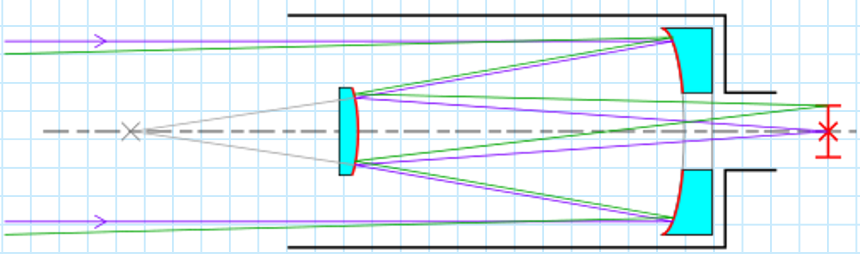
- TLE stands for Two-Line Element. It is a list of "orbital elements" of an object orbiting earth
- "Orbital elements" - eccentricity, semimajor axis, inclination, longitude of ascending node, argument of periapsis, true anomaly
  - https://en.wikipedia.org/wiki/Orbital_elements
- Essentially gives information about the satellite and its orbit

UART

Universal asynchronous receiver / transmitter
Serial communication protocol

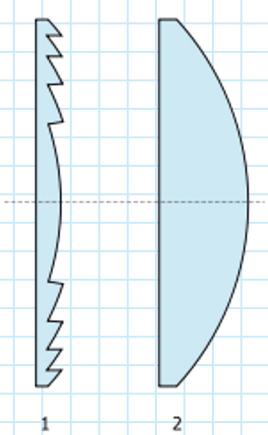
Cassegrain Reflector

Used in the image capturing process



Refers to this setup.
https://en.wikipedia.org/wiki/Cassegrain_reflector

Fresnel Lens



https://en.wikipedia.org/wiki/Fresnel_lens

Allows for a lens to be made using less material by placing the various angles on the same plane

UHF vs S-Band
https://solarsystem.nasa.gov/basics/chapter6-3/

Board Level Camera
https://www.lumenera.com/media/wysiwyg/resources/documents/datasheets/industrial/Board_Level_Cameras_Data_Sheet.pdf

https://docs.google.com/spreadsheets/d/1J8YjafzkXoyC-gaV5bLQfIQjLdRwd8SVoV2DYM6fr5k/edit?usp=sharing

Name	Size	Resolution	Pixel Size	Power	Shutter	Digitization	Connectivity	Link
CMOSIS CMV4000	26x26mm	2048x2048	5.5	1.6W	Global	8, 10, 12*	USB3	https://www.ximea.com/en/
Sony IMX250	26x26x33 mm	2464x2056	3.45	2.85W	Global	8, 10, 12	USB3	https://www.ximea.com/en/
Aptina MT9P031	13x13x3mm	2592x1944	2.2	>1W	Rolling	12	USB	https://www.ximea.com/en/
DFM 37UX250-ML	36x36x15mm	2448x2048	3.45	2.05W	Global	12	USB3.1	https://www.theimagingsour
DFM 37UX264-ML	36x36x15mm	2448x2048	3.45	1.8W	Global	12	USB3.1	https://www.theimagingsour
DFM 37UX178-ML	30x30x15	3072x2048	2.4	1.6W	Global	12	USB3.1	https://www.theimagingsour
LT-C2420B	40.4x40.4x17.1mm	2464x2056	3.45			8, 12	USB3.1	https://www.phase1vision.c

The first one I found seems to be the best. It has a small form, large pixels and low power.
https://www.ximea.com/en/products/xilab-application-specific-custom-oem/board-level-cameras-whole-range/cmosis-cmv4000-usb3-color-camera

Computers
https://jaycarlson.net/embedded-linux/

So you want to build an...

MICROCHIP High Performance: SAMS70/E70/V7x Series

SAMS70/E70/V7x Series

Feature	SAMS70	SAME70	SAMV70	SAMV71
Frequency	300 MHz			
Flash	512 KB/1 MB/2 MB	512 KB/1 MB/2 MB	512 KB/1 MB	512 KB/1/MB/2 MB
SRAM	256 KB/384 KB/384 KB	256 KB/384 KB/384 KB	256 KB/384 KB	256 KB/384 KB/384 KB
Backup SRAM	1 KB			
Ext. Bus Interface	16-bit (SDRAM, SRAM)			
Ethernet 1588 (MAC)	–	10/100 Mbps	–	10/100 Mbps
CAN FD	–	2	2	2
MediaLB®	–		Yes	
Hi-Speed USB	1			
Automotive Qualified	–		Yes	
Camera interface	1			
QSPI	1			
HSMCI/SDIO/eMMC	1x HS			
USART or SPI/UART	5/3			
SPI/I²C/SSC (I²S/TDM)	2/3/1			
12-bit ADC	2x 12-ch 2 Msps			

HSMCI/SDIO/eMMC	1x HS
USART or SPI/UART	5/3
SPI/I²C/SSC (I²S/TDM)	2/3/1
12-bit ADC	2x 12-ch 2 Msps
12-bit DAC	2-ch 2 Msps
Timers/PWM	12/8
Crypto	TRNG, AES 256, SHA 1/256
Pin Count	64–144
Package	QFN, QFP, BGA

High Performance
- Arm Cortex-M7: 300 MHz, 1500 CoreMark™
- Single- and double-precision hardware Floating Point Unit (FPU)
- 16 KB+ 16 KB of I&D cache with ECC
- Execution in place from on-chip Flash NVM connected to QSPI and EBI
- Multi-port SRAM minimizing latency
- User-configurable SRAM and TCM size

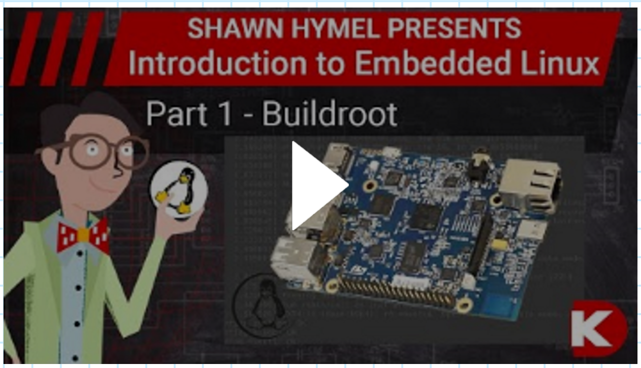
Advanced Analog Front-End (AFE)
- Dual S&H, 12-bit ADC and 16-bit hardware averaging
- Differential input, programmable gain
- Automatic gain and offset error correction
- DMA support, hardware and software trigger

Features
- Hi-Speed USB host/device with integrated PHY
- Memory integrity check monitor
- CMOS camera interface
- Ethernet and dual CAN on SAME70 and SAMV71
- Sleepwalking on UART and I²C
- Event system

Temperature Options
- −40°C to +105°C
- AEC-Q100, −40°C to +105°C (Grade 2)

6 www.microchip.com/32bit

[Introduction to Embedded Linux Part 1 - Buildroot | Digi-Key Electronics]



Set up Platform IO for current computer

I installed the package however the device is not being recognized.

It does not seem to be an issue with the package, as windows is also not detecting the device.

Leave 4:00 PM

01/06/23

Time: 8h

Thursday, June 1, 2023 8:01 AM

I Feel like I need to better review the peripherals going into the MCU/MPU before doing more research on possible boards

Components that need to interface with MCU/MPU
- S-band transmitter
- Sun sensors
- UHF transeiver
- IMU
- Magnetorquers H-bridges
- Reaction wheel motor drivers
-

- 12 Analog pins for sun sensors
- USB 3 for camera
- SPI for S-band transmitter
- The UHF is not defined in the Subsystem Design Document
- I2C/SPI for IMU
- Assume 2 Digital output pins / magnetorquer
- 6 PWM for Magnetorquers
- ~~Motor driver requirements not known~~
- 1 PWM & 4 digital per motor driver
- 2 PWM & 8 digital for motor drivers
- UHF uses UART/I2C

MCU/MPU requirements
- 12 channel 12-bit ADC
- Usb 3
- 1 SPI
- 2 I2C
- 8 PWM
- 8 digital

The biggest issue here is the USB 3 capability, which is primarily from me searching for a USB 3 camera

S-Band

	Default Configuration	Optional Configuration
Tx Frequency Band	2.200-2.290 GHz	2.100-2.500 GHz
Data rate (Net Payload Data)	1 Mbps	0.4 ... 1.6 Mbps
Tx RF Bandwidth	1.3 MHz	
RF Power Output (w/o aerial)	+27 dBm	Up to +30 dBm (adjustable)
Tx Modulation Scheme	DQPSK	
FEC scheme	TURBO code, r = 0.489	Settable rate r = 0.22 ... 0.79
RF Connector Type	SMA male 50 Ω, cable	Customer specific
Data Interfaces	3-wire (SPI, data) UART (telemetry)	UART (telecommand) UART (data)
Connector Type	SMC 1.27mm female 12-pins, cable (data, signaling) 2x 0.5mm² wires (power)	Customer specific
DC supply	+3.3 ... 5.0 V	
DC Power Consumption	<5 W	Depends upon RF power
Mechanical Dimensions	95 x 46 x 15 mm³	
Mass	100 grams (incl. housing & cables)	
Temperature Range	-20°C ... +50°C (operational)	
Technology Readiness Level (TRL)	9	
Case	Passivated aluminum	

11:AM break for meeting
Back 12 pm

Questions

- What specifically is wrong with the current MCU system.
- The TRR document outlines how a software update could be sent from the ground to the satellite. How does the ground control the microcontroller to send the update? has this been developed
- Overall what has actually been developed for the Computer system? The TRR reads like there is a plan for all this but I cant see any documentation on the details
- ~~From my understanding, PWM is needed to control the speed of the motors and the current through the magnetorquers, which would require 8 PWM pins. From what I can see the IOnuverons board only has 2 PWM channels Is there more information of their design. If the magnetorquers are being controlled as -5V, 0V, +5V the numbers will add up.~~

After reviewing the b-dot algorithm I believe the magnetorquers are controlled binary as + or - which means the H-bridge only needs digital IO not PWM

MPU Search

I am not finding very many MPU options that have USB 3 for the camera. The ones that I am finding that are capable are generally just the chip with no peripherals such as ADCs and PWM which will be needed for the sun sensors and motors.

One options would be to have a MCU with built in ADCs and PWM. This would act as just the controller for the hardware while the MPU did the processing and accepted user commands.

This MPU has USB3 capability, and some communication interfaces
https://www.ti.com/lit/ds/symlink/am5k2e04.pdf?ts=1685621479899&ref_url=https%253A%252F%252Fwww.google.com%252F

However there is no ADC, PWM and the chip does not have any development board.

This TI series of boards almost has everything
https://www.ti.com/lit/ds/symlink/am6421.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1685622181845&ref_url=https%253A%252F%252Fwww.mouser.ca%252F

The main issue I see is that it only had 1 ADC multiplexed to 8 channels.

It may be too complicated but there may be a way to do the current sums and differences in hardware to only need 2 analog channels per sensor as opposed to 4

Grant just shared the CubeSat Basecamps with me, I will need to review those.

PDF
Motor Driver...

Motor Driver Datasheet

maxon motor
Operating Instructions 1-Q-EC Amplifier DEC Module 24/2

5 Functional Description of Inputs and Outputs

5.1 Inputs

5.1.1 Speed range and mode selection with «DigIN1» und «DigIN2»

2 digital
1 PWM

The digital inputs «DigIN1» [11] and «DigIN2» [12] determine both, the operation mode (digital speed controller or digital speed actuator) and the speed range in speed set value mode.

		Motor type		
DigIN1	**DigIN2**	**1 pole pair**	**4 pole pair**	**8 pole pair**
0	**0**	Open loop speed control, 0...100 % PWM depending on the «Set value speed» input voltage		
1	**0**	500...5 000 rpm	125...1 250 rpm	62...625 rpm
0	**1**	500...20 000 rpm	125... 5 000 rpm	62...2 500 rpm
1	**1**	500...80 000 rpm	125...20 000 rpm	62...10 000 rpm

Please note
⇨ If the signal level of the digital inputs DigIN1 [11] and DigIN2 [12] are changed, the new levels are adopted by a disable-enable procedure.

If the input «DigIN» is not connected (floating) or a voltage higher than 2.4 V is applied, the input is active.

Logic 1	Input not connected (floating) Input voltage > 2.4 V	Input active

If the input «DigIN» is set to ground potential or a voltage smaller than 0.8 V is applied, the digital input is inactive

Logic 0	Input set to Gnd Input voltage < 0.8 V	Input inactive

The inputs «DigIN1»and «DigIN2» are protected against over voltage.

Digital input 1	Pin number [11] «DigIN1»
Digital input 2	Pin number [12] «DigIN2»
Input voltage range	0 ... +5 V
Input impedance	15 kΩ pull-up resistor against 5 V
Continuous over voltage protection	-28 ... +28 V

maxon motor
1-Q-EC Amplifier DEC Module 24/2 Operating Instructions

5.1.2 Set value «Set value speed»

At the «Set value speed» input [17] the external analogue set value and hence the rotational speed of the motor shaft is predetermined.
By adjusting the signal levels on digital inputs «DigIN1 [11]» and «DigIN2 [12]» the speed range can be set in advance.

		Motor type		
DigIN1	**DigIN2**	**1 pole pair**	**4 pole pair**	**8 pole pair**
0	**0**	Open loop speed control, 0...100 % PWM depending on the «Set value speed» input voltage		
1	**0**	500...5 000 rpm	125...1 250 rpm	62...625 rpm
0	**1**	500...20 000 rpm	125...5 000 rpm	62...2 500 rpm
1	**1**	500...80 000 rpm	125...20 000 rpm	62...10 000 rpm

Note
⇨ If the signal level of the digital inputs DigIN1 [11] and DigIN2 [12] are changed, the new levels are adopted by a disable-enable procedure.

Set value voltage	Description
0 V ... 0.1 V	Operation at a minimum speed
0.1 V ... 5.0 V	Linear speed adjustment

The actual speed value is calculated according the following formula:

Known values
⇨ Minimum speed (see table above) n_{min} [rpm]
⇨ Maximum speed (see table above) n_{max} [rpm]
⇨ Set value voltage V_{set} [V] respectively speed n [rpm]

Sought value
⇨ speed n [rpm]
Solution

$$n = \left[\frac{V_{set} - 0.1[V]}{4.9[V]} \cdot (n_{\max} - n_{\min})\right] + n_{\min}$$

Sought value
⇨ Set value voltage V_{set} [V]
Solution

$$V_{set} = \left(\frac{n - n_{\min}}{n_{\max} - n_{\min}} \cdot 4.9[V]\right) + 0.1[V]$$

The «Set value speed» input is protected against over voltage.

Set value speed input	Pin number [17] «Set value speed»
Input voltage range	0 ... +5 V (referenced to Gnd)
Resolution	1024 steps (4.88 mV)
Input impedance	107 kΩ (in range 0 ... +5 V)
Continuous over voltage protection	-28 ... +28 V

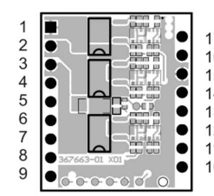
⚠ **The change rate of the set value signal is limited internally with a ramp function. It nominally takes 1 s to reach the maximum speed for the selected speed range. This time can be shortened proportionally by**

3 Pin assignment DEC Module 24/2

Top view

1 PWM
4 Digital pins

• Motor
• MCU
• Other



3.1 Pin assignment

Pin	Signal	Description
1	W1	Motor winding 1 •
2	W2	Motor winding 2 •
3	W3	Motor winding 3 •
4	$+V_{CC}$	Supply voltage 8...24 VDC •
5	Gnd	Ground •
6	V_{CC} Hall	+5 VDC output voltage •
7	H1	Hall sensor 1 •
8	H2	Hall sensor 2 •
9	H3	Hall sensor 3 •

Pin	Signal	Description
17	Set value speed	Set value speed input • PWM
16	Set current limit	Set current limit input •
15	Gnd	Ground •
14	Direction	Direction input • Digital
13	Enable	Enable input • Digital
12	DigIN2	Digital input 2 • Digital
11	DigIN1	Digital input 1 • Digital
10	Ready	Status indication output •

Continuous over voltage protection	-28 ... +28 V

Set value speed input	Pin number [17] «Set value speed»
Input voltage range	0 ... +5 V (referenced to Gnd)
Resolution	1024 steps (4.88 mV)
Input impedance	107 kΩ (in range 0 ... +5 V)
Continuous over voltage protection	-28 ... +28 V

The change rate of the set value signal is limited internally with a ramp function. It nominally takes 1 s to reach the maximum speed for the selected speed range. This time can be shortened proportionally by defining smaller set value increments.

7	H1	Hall sensor 1	12	DigIN2	Digital input 2 Digital
8	H2	Hall sensor 2	11	DigIN1	Digital input 1 Digital
9	H3	Hall sensor 3	10	Ready	Status indication output

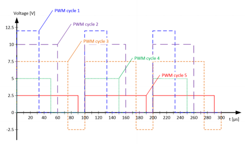
Adjusting set values via PWM control

Instead of an analog voltage, a PWM signal with a fixed frequency and amplitude can be used to adjust the speed set value.
The desired change in the set value is achieved by variation of the duty cycle in the range 0…100%. Both the amplitude and the duty cycle have an influence on the resulting speed. The mean value of the applied PWM voltage corresponds to the analog input signal for the speed set value.

Nominal value amplitude PWM set value	0…5 V
Max. value amplitude PWM set value	-28…+28 V
Frequency range PWM set value	500 Hz…20 kHz
Modulation PWM set value	0…100%
Continuous overvoltage protection	-28…+28 V

Examples:
motor type: 1 pole pair
speed range: 500...20'000 min^{-1}

$$n = \left[\frac{V_{set} - 0.1[V]}{4.9[V]} \cdot (n_{\max} - n_{\min})\right] + n_{\min}$$



PWM cycle 1: 33 % PWM @ 0 V ... 12 V → 4.0 V → 16'020 rpm

PWM cycle 2: 60 % PWM @ 0 V ... 10 V → 6 V limited to 5 V (max. set value voltage) → 20'000 rpm

PWM cycle 3: 75 % PWM @ -2.5 V ... 7.5 V → 5.0 V → 20'000 rpm

PWM cycle 4: 50 % PWM @ 0 V ... 5 V → 2.5 V → 10'051 rpm

PWM cycle 5: 90 % PWM @ 0 V ... 2.5 V → 2.25 V → 9'056 rpm

5.1.3 «Enable»

digital

The «Enable» input enables or disables the power stage.

If a voltage higher than 2.4 V is applied to the «Enable» input, the amplifier is activated (Enable). A speed ramp will be performed during acceleration.

Enable	Input voltage > 2.4 V	Motor shaft running

If the input is not connected (floating) or ground potential is applied to the «Enable» input, the power stage is high impedant and the motor shaft freewheels and slows down (Disable).

Disable	Input not connected (floating) Input set to Gnd Input volatge < 0.8 V	Power stage switched off

The «Enable» input is protected against overvoltage.

Enable	Pin number [13] «Enable»
Input voltage range	0 ... +5 V
Input impedance	100 kΩ (in range 0 ... +5 V)
Continuous over voltage protection	-28 ... +28 V
Delay time	max. 20 ms

Note
⇨ If the signal level of the digital inputs DigIN1 [11] and DigIN2 [12] are changed, the new levels are adopted by a disable-enable procedure.

5.1.4 «Direction»

digital

The «Direction» input determines the rotational direction of the motor shaft. When the level changes, the motor shaft slows down with a ramp to standstill, and accelerates with a speed ramp in the opposite direction, until the nominal speed is reached again.

If the input is not connected (floating) or ground potential is applied to the «Direction» input, the motor shaft runs clockwise (CW).

CW	Input not connected (floating) Input set to Gnd Input voltage < 0.8 V	Clockwise (CW)

If a voltage higher than 2.4 V is applied to the «Direction» input, the motor shaft runs counter-clockwise (CCW).

CCW	Input voltage > 2.4 V	Counter-clockwise (CCW)

The «Direction» input is protected against overvoltage.

Direction	Pin number [14] «Direction»
Input voltage range	0 ... +5 V
Input impedance	100 kΩ (in range 0 ... +5 V)
Continuous over voltage protection	-28 ... +28 V

5.1.5 «Set current limit»

Resistor Value (Assume Const.)

The «Set current limit» inputs is used for setting the contiunuous output current limitation in the range of 0.5...3 A.
The current applied at the input «Set current limit» will stay available for an indefinite period of time.

Note
⇨ The limiting value should be below the rated motor current (max. continuous current) as shown on the motor data sheet (corresponds to line 6 in maxon catalogue).

Set value current	Pin number [16] «Set current limit»
Referenced to Ground	Pin number [15] «Gnd»

To parameterize the preferred current limiting value, an external resistor (at least 62.5 mW) between current limiting input «Set current limit» Pin [16] and ground «Gnd» Pin [15] must be added.

Current limit value	Resistance value (E12 series)
3.0 A	input floating
2.5 A	47 kΩ
2.0A	10 kΩ
1.5 A	4.7 kΩ
1.0 A	2.2 kΩ
0.5 A	470 Ω

Note
⇨ Under unfavourable circumstances the actual motor peak current can not be limited to the set current limit in all cases.
Unfavourable circumstances are given, if the current limit value is set lower than 1.5 A, the supply voltage is higher than 15 V an the terminal inductance is smaller than 0.3 mH at the same time.

5.1.6 «Hall sensor 1», «Hall sensor 2», «Hall sensor 3»

Connected to motor

Hall sensors are needed for detecting rotor position and actual speed.

The Hall sensor inputs are protected against overvoltage.

Hall sensor 1	Pin number [7] «Hall sensor 1»
Hall sensor 2	Pin number [8] «Hall sensor 2»
Hall sensor 3	Pin number [9] «Hall sensor 3»
Input voltage range	0 ... +5 V
Input impedance	10 kΩ pull-up resistor to 5 V
Voltage level «low»	max. 0.8 V
Voltage level «high»	min. 2.4 V
Continuous over voltage protection	-28 ... +28 V

Suitable for Hall sensor IC's with Schmitt-Trigger behavior and open collector outputs.

5.2 Outputs

5.2.1 +5 VDC output voltage «V_{cc} Hall»

An internal auxiliary voltage of +5 VDC is provided for:
⇨ Hall sensor supply voltage «V_{cc} Hall»
⇨ For external set value potentiometer (recommended value: 10 kΩ)
⇨ Gating the signals: «Enable» and «Direction»
The output is thermal overload protected against short circuit.

+5 VDC output voltage	Pin number [6] «V_{cc} Hall»
Referenced to Ground	Pin number [5] «Gnd»
Output voltage	+5 VDC ± 5 %
Max. output current	35 mA

5.2.2 Status indication «Ready»

The «Ready» output can be used to report the state of operational readiness or a fault condition to a master control unit.

In normal cases (no fault) the output is switched to 5V.

Ready (no fault)	5 V

In case of a fault the output is switched to Ground.

Fault (not ready)	0 V (Gnd)

Possible reason for a fault message:
⇨ **Undervoltage**
Fault message occurs in case supply voltage $+V_{cc}$ < 6.5 VDC.
To reset the fault condition the amplifier must be disabled and the supply voltage $+V_{cc}$ must be higher than 6.5 VDC.
⇨ **Overvoltage**
Fault message occurs in case supply voltage $+V_{cc}$ > 30 VDC.
To reset the fault condition the amplifier must be disabled and the supply voltage $+V_{cc}$ must be lower than 29 VDC.
⇨ **Thermal overload**
Fault message occurs in case power stage temperature exceeds > 95°C.
To reset the fault condition the amplifier must be disabled and the power stage temperature must fall below 75°C
⇨ **Invalid Hall sensor signals**
The amplifier recognizes invalid conditions in the Hall sensor inputs during the power-up. To reset the fault condition the amplifier must be disabled and the Hall sensors must be wired correctly.

The output «Ready» is protected against short circuit.

Status indication	Pin number [12] «Ready»
Output voltage range	0 ... +5 V
Output resistance	10 kΩ

Computer

This is a complete development board with all the peripherals except for 12 ADC channels (8)
https://www.phytec.com/product/phyboard-am64x-development-kit/

I have found that USB 3 is my major limitation. I don't know what makes USB 3 different from USB 2 aside from the fact that its faster

If I can use the USB3 camera in a USB2 port then my options will increase dramatically
https://www.cmd-ltd.com/advice-centre/usb-chargers-and-power-modules/usb-and-power-module-product-help/usb-compatibility/#:~:text=with%20USB%202.0%3F-,%E2%80%94%20A%20guide%20to%20USB%20versions%20and%20compatibility,speeds%20of%20USB%203.0%20technology.

From this, I should be able to us a USB 3 device in a USB 2 port.

The main difference is the data transfer rate. USB 3 has a data transfer rate of 5Gbit/s while USB has a rate of 480Mbit/s

This would mean that USB 3 is approximately 10.4x faster than USB 2. Assuming that the data transfer rate is the limiting factor in the cameras FPS rating, this would mean that the 90fps 3.0 camera would be ~9fps on a USB 2 port. This may be sufficient, as the plan is not to record video with this camera, only images.

Another issue is finding a MPU with 12 ADC channels. There are options with 7, but the 12 is harder to find.

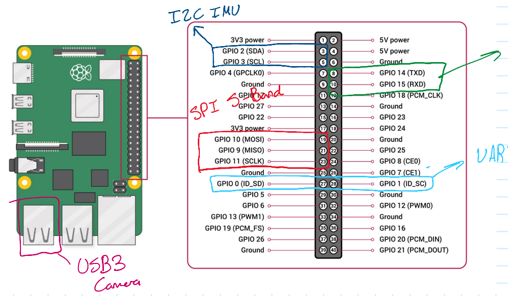
The main thing about the original design that I do not understand is interfacing with the camera. If it isn't a USB connector, I don't really know how to interface with a camera. I want to get a good understanding of the original design to have a better understanding of what needs to be changed, and have a discussion about this with Nick.

Raspberry Pi 4 has the required USB3 ports, sufficient IO pins, SPI, I2C but no PWN or ADC
A simple microcontroller could be connected that would actually interact with the hardware, and be controlled by the main computer raspberry pi
Its not ideal that the Raspberry pi has so many other features that arent being used, but it could be an option.

One thing that I am seeing is that there may not be serial communication with the pi.

If someone wanted to communicate with the PI from the ground, I do not know how that works.

I am reading that the PI does have UART Serial commincation over some of its digital pins

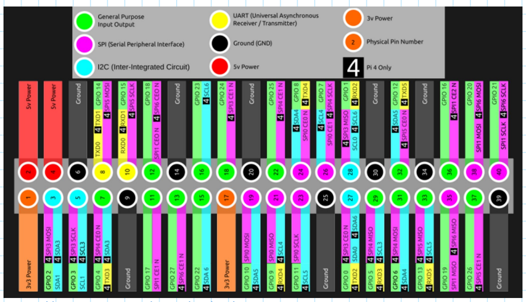


MCU interfaces with
- magnetorquer H-bridges - 6 PWM
- reaction wheel motor drives - 2PWM - 8 digital
- Sun sensor outputs - 12 Analog

https://elinux.org/RPi_Serial_Connection
https://medium.com/@sarala.saraswati/connecting-to-your-raspberry-pi-console-via-the-serial-cable-44d7df95f03e

The UHF transmitter could be connected to that interface to communicate with the satellite.

The S-Band then needs a SPI interface, and a I2C for the IMU



https://hackaday.com/2022/02/01/did-you-know-that-the-raspberry-pi-4-has-more-spi-i2c-uart-ports/

I am having trouble finding a board with the 8 PWM signals to control the two motors and 3 magnetorquers. Looking into the board that the team had previously selected, the IONeuron board, only 2 PWM generators are available on the board. I am not sure how they are controlling everything.

I was thinking that the magnetorquers would need a PWM on each direction of the H-bridge. This would mean 2/magnetorquer or 6 total. However the direction pins could be set using a digital pin and the supply power to the H-bridge could be adjusted using PWM on a FET. This would reduce it to 1 PWM per magnetorquer which would reduce the total PWM from 8 to 5.

This is still more than the 2 on the IONeuron board...

The motors are being controlled using PWM
I can see that in one of the test code files.

```
void loop() {
  if(Count == 0){
    analogWrite(SET_SPEED, 255*Duty_Cycle/100);
  }
```

Maybe the magnetorquers don't need adjustable currents?
To get fine control it would definitely be needed, however maybe the plan was to treat as a + 0 - control as opposed to analog.

Revisiting the website on b-dot algorithm I read yesterday, I can see that it is saying it is controlled as a binary value and not an analog. This means that instead of the 8PWM channels and 8 digital channels, I need 2 PWM and 14 digital. This will be much easier to find.

https://www.digikey.ca/en/products/detail/microchip-technology/PIC24FJ256GA702-I-SP/6562000
This microcontroller should have the required PWN and ADC pins
It also can connect with UART

Jakob informed me of the SCADA 3rd year design project document and that it would likely be realivent to the control system

I began reviewing that however it was very confusing

Leave - 5:00

02/06/23

Friday, June 2, 2023 8:15 AM

Time: 7.5h

Arrive 8:15

Yesterday I found that combining the Raspberry Pi 4 with a microcontroller would give required interfaces. I know that the Raspberry Pi has a lot of other features that are not being utilized. It would likely be more power efficient to find a simpler board, however google searches do not seem to differentiate MPU and MCU very well.

I also still want to know what is the main issue with the current design. From what I see, the main issue would be interfacing another camera. Nick had suggested finding a USB3 camera and a MPU with USB3, but it seems kind of overkill to go with the full MPU system just to have USB3 for a camera. I know he also mentioned wanting Linux, but I'm not sure what features would be needed that couldn't be programmed onto a MCU. For example, he had mentioned being able to give the satellite instructions in a terminal. Really the MCU could be programmed to have a Serial terminal that accepted preprogrammed commands.

I am also realizing that my system requirements outlined yesterday have an error. I did not consider pins for monitoring the batteries, which is something that Nick had mentioned. There may also be additional analog inputs required that were not specifically outlined in the document.

Questions

- Is there a system for monitering the batteries. If so, what outputs does it have?
- Regarding upgrading to a MPU with a full OS as opposed to the MCU's, I have concerns regarding the power. The original design had an average net power of 0.7W.
- What was the FPGA doing in the original design? Was there actually a plan on what specifically it needed to do and roughly how that would work? Cause I cant find any details aside from "The camera will attach to the FPGA"

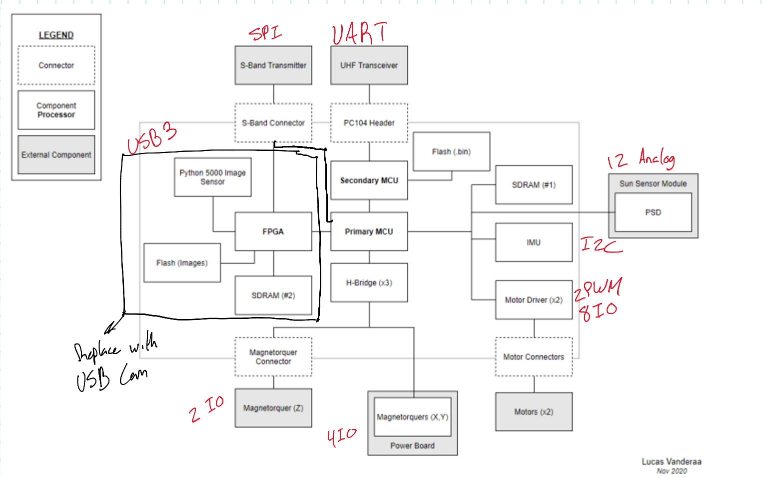


Figure 8.8: OBC Block Diagram

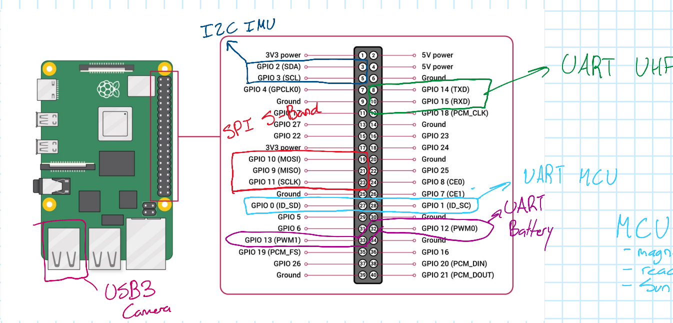
This is supposed to be the block diagram for the computer.

This does not show anything monitoring the power and communicating that to the computer.

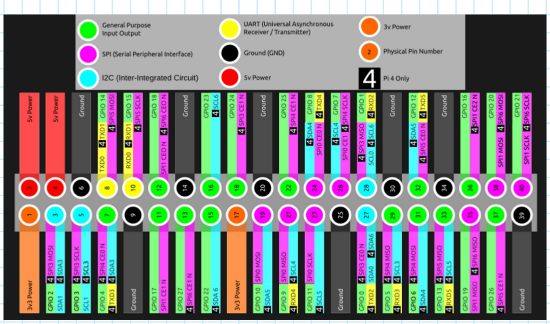
Does this system exist? It will be important when defining the requirements for the microcontroller.
The controller I found yesterday has 14 ADC channels which means that there would be 2 space channels to monitor the battery. IF that is enough...

I revisited the document and found that the battery is monitored by the BQ79606A-Q1 chip which communicates using UART.

From Yesterday, the battery monitor can be added to pin 31 and 33



MCU interfaces with
- magnetorquer H-bridges - 6 PWM
- reaction wheel motor drives - 2PWM - 8 digital
- Sun sensor outputs - 12 Analog



9AM break

9:30 back

As I mentioned earlier, using the raspberry pi may create some power concerns as it is much more advanced than the FGPA and microcontrollers

The raspberry Pi uses a 5V 3A power supply (15W) which is a lot. I'm sure that this amount of power is not used all of the time.

According to this website, the Pi uses around 3W when idling and 4.5W loading a LXDE OS. Im sure that the OS would also make a significant difference when looking into the power requirements.

I found a Linux disto called Tiny Core Linux. This appears to be a very light weight OS for raspberry pi.
https://www.makeuseof.com/tag/lightweight-operating-systems-raspberry-pi/
http://tinycorelinux.net/downloads.html

If the method of a raspberry pi is persued, it would be good to do some initial testing to measure the power draw of the OS while idling.

Additionally, it would be good to look into a power breakdown of the system to see how much power can be used by the computer.

Table 5-1: A power budget summary for each of the subsystems broken down into the operational modes.

Operational Mode	Active Mode	Standby Mode	Power Savings Mode
Mode Time (min)	16.7	1392.1	77.3
Total Time (min) for 16 Orbits	1486.1	1486.1	1486.1
Subsystem	**Energy Consumption [Wh]**	**Energy Consumption [Wh]**	**Energy Consumption [Wh]**
Structure + Payload	0.111	1.159	0.064
Communications	0.495	5.797	0.322
Command + Control	1.712	25.100	0.519
Power	0.111	9.275	0.515
20% Contingency	0.500	8.266	0.284
Total Energy Consumed (Wh)	2.930	49.596	1.704
Average Modal Power Consumption (W)	**10.526**	**2.138**	**1.323**
Average Power Consumption (W)		2.190	
Average Power Generated (W)		2.977	
Average Net Power (W)		**0.787**	

To better understand this, I need to know a break down of the 'Command + Control' subsystem. That would likely also contain the motors and magnetorquers which are not needed for this. Also I would like to know how they determined the power requirements in the various modes.

"CCP-SN1-CDR-0120 Power Budget" Excel sheet apparently has a full breakdown of that table.

This budget it based on a MATLAB simulation that determined the values for time spent in active mode. Based on 16 Orbits.

Specific Power Consumption In Active Mode			16.7		1485.2
Component	**Rate [W]**		**Time [min]**		**Draw [Wh]**
	Active	*Passive*	*Active*	*Passive*	
Structure + Payload					
Image Sensor	1.45	0.05	4.175	12.525	0.111
Communications					
UHF Transceiver	1.363	0.15	4.175	12.525	0.126
S Band Transmitter	5	0.1	4.175	12.525	0.369
Command + Control					
On-Board Computing	-	-	16.7	0	
Primary MCU	1.08	0	16.7	0	0.301
Primary MCU Flash			0	0	0
Secondary MCU	0.4	0	16	0	0.107
Flash for Secondary MCU			0	0	0
FPGA	1	0	4.175	0	0.070
FPGA RAM			0	0	0
FPGA Flash			0	0	0
Reaction Wheels	1.5	0	16.7	0	0.418
Motor		0			
Motor Driver		0			
Magnetorquers	2.1	0	16.7	0	0.585
IMU	0.0056	0.0025	4.175	12.525	0.001
PSD	0	0	0	0	0
Power					
Power System	0.2	0	16.7	0	0.056
Battery Heaters	0.8	0	16.7	0	0.223
20% Contingency					0.473
Total					**2.837**
Average Power Consumption In Active Mode (W)					10.194

Specific Power Consumption in Standby			1391.2		1485.2
Component	**Rate [W]**		**Time [min]**		**Draw [Wh]**
	Active	*Passive*	*Active*	*Passive*	
Structure + Payload					
Image Sensor	1.45	0.05	0	1391.2	1.159
Communications					
UHF Transceiver	1.4	0.15	0	1391.2	3.478
S Band Transmitter	5	0.1	0	1391.2	2.319
Command + Control					
On-Board Computing	-	-	1391.2	0	
Primary MCU	1.08	0	1391.2	0	25.042
Primary MCU Flash			0	0	0
Secondary MCU	0.4	0	0	1391.2	0
Flash for Secondary MCU			0	0	0
FPGA	1	0	0	0	0
FPGA RAM			0	0	0
FPGA Flash			0	0	0
Reaction Wheels	1.5	0	0	0	0
Motor		0			
Motor Driver		0			
Magnetorquers	2.1	0	0	0	0
IMU	0.0056	0.0025	0	1391.2	0.058
PSD	0	0	0	0	0
Power					
Power System	0.2	0	1391.2	0	4.637
Battery Heaters	0.8	0	1391.2	0	18.549
20% Contingency					11.048
Total					**66.291**
Average Power Consumption In Standby Mode (W)					2.859

Specific Power Consumption in Power Saving Mode			77.3		1485.2
Component	**Rate [W]**		**Time [min]**		**Draw [Wh]**
	Active	*Passive*	*Active*	*Passive*	
Structure + Payload					
Image Sensor	1.45	0.05	0	77.3	0.064
Communications					
UHF Transceiver	1.4	0.15	0	77.3	0.193
S Band Transmitter	5	0.1	0	77.3	0.129
Command + Control					
On-Board Computing	-	-	77.3	0	
Primary MCU	1.08	0	0	0	0
Primary MCU Flash			0	0	0
Secondary MCU	0.4	0	77.3	0	0.515
Flash for Secondary MCU			0	0	0
FPGA	1	0	0	0	0
FPGA RAM			0	0	0
FPGA Flash			0	0	0
Reaction Wheels	1.5	0	0	0	0
Motor		0			
Motor Driver		0			
Magnetorquers	2.1	0	0	0	0
IMU	0.0056	0.0025	0	77.3	0.003
PSD	0	0	0	0	0
Power					
Power System	0.2	0	77.3	0	0.258
Battery Heaters	0.8	0	77.3	0	1.031
20% Contingency					0.439
Total					**2.632**
Average Power Consumption In PSM (W)					2.043

In this Form, If I can find a passive and active power for the PI, MCU and Image sensor, I can update those values and it should calculate the rest.
I know the Camera is rated for 1.6W active. I do not know what the passive power draw would be.

If I fill in the values of 4.5W for active mode, 3W for stand by, and 3W for power saving, the average power drawn is nearly double what is being generated.

Specific Power Consumption In Active Mode			16.7		1485.2
Component	**Rate [W]**		**Time [min]**		**Draw [Wh]**
	Active	*Passive*	*Active*	*Passive*	
Structure + Payload					
Image Sensor	1.6	0.05	4.175	12.525	0.122
Communications					
UHF Transceiver	1.363	0.15	4.175	12.525	0.126
S Band Transmitter	5	0.1	4.175	12.525	0.369
Command + Control					
On-Board Computing	-	-	16.7	0	
Primary MCU	4.5	0	16.7	0	1.253
Primary MCU Flash			0	0	0
Secondary MCU	0	0	16	0	0.000
Flash for Secondary MCU			0	0	0
FPGA		0	4.175	0	0.000
FPGA RAM			0	0	0
FPGA Flash			0	0	0
Reaction Wheels	1.5	0	16.7	0	0.418
Motor		0			
Motor Driver		0			
Magnetorquers	2.1	0	16.7	0	0.585
IMU	0.0056	0.0025	4.175	12.525	0.001
PSD	0	0	0	0	0
Power					
Power System	0.2	0	16.7	0	0.056
Battery Heaters	0.8	0	16.7	0	0.223
20% Contingency					0.630
Total					**3.781**
Average Power Consumption In Active Mode (W)					13.583

Specific Power Consumption in Standby			1391.2		1485.2
Component	**Rate [W]**		**Time [min]**		**Draw [Wh]**
	Active	*Passive*	*Active*	*Passive*	
Structure + Payload					
Image Sensor	1.6	0.05	0	1391.2	1.159
Communications					
UHF Transceiver	1.4	0.15	0	1391.2	3.478
S Band Transmitter	5	0.1	0	1391.2	2.319
Command + Control					
On-Board Computing	-	-	1391.2	0	
Primary MCU	3	0	1391.2	0	69.560
Primary MCU Flash			0	0	0
Secondary MCU	0	0	0	1391.2	0
Flash for Secondary MCU			0	0	0
FPGA	1	0	0	0	0
FPGA RAM			0	0	0
FPGA Flash			0	0	0
Reaction Wheels	1.5	0	0	0	0
Motor		0			
Motor Driver		0			
Magnetorquers	2.1	0	0	0	0
IMU	0.0056	0.0025	0	1391.2	0.058
PSD	0	0	0	0	0
Power					
Power System	0.2	0	1391.2	0	4.637
Battery Heaters	0.8	0	1391.2	0	18.549
20% Contingency					19.952
Total					**119.713**
Average Power Consumption In Standby Mode (W)					5.163

Specific Power Consumption in Power Saving Mode			77.3		1485.2
Component	**Rate [W]**		**Time [min]**		**Draw [Wh]**
	Active	*Passive*	*Active*	*Passive*	
Structure + Payload					
Image Sensor	1.6	0.05	0	77.3	0.064
Communications					
UHF Transceiver	1.4	0.15	0	77.3	0.193
S Band Transmitter	5	0.1	0	77.3	0.129
Command + Control					
On-Board Computing	-	-	77.3	0	
Primary MCU	3	0	77.3	0	0
Primary MCU Flash			0	0	0
Secondary MCU	0	0		0	0.000
Flash for Secondary MCU			0	0	0
FPGA	1	0	0	0	0
FPGA RAM			0	0	0
FPGA Flash			0	0	0
Reaction Wheels	1.5	0	0	0	0
Motor		0			
Motor Driver		0			
Magnetorquers	2.1	0	0	0	0
IMU	0.0056	0.0025	0	77.3	0.003
PSD	0	0	0	0	0
Power					
Power System	0.2	0	77.3	0	0.258
Battery Heaters	0.8	0	77.3	0	1.031
20% Contingency					0.336
Total					**2.014**
Average Power Consumption In PSM (W)					1.563

Average Power 5.0703

Battery Heaters	0.8	0	16.7	0	0.223	**Battery Heaters**	0.8	0	1391.2	0	18.549	**Battery Heaters**	0.8	0	77.3	0	1.031
20% Contingency					0.630	20% Contingency					19.952	20% Contingency					0.336
Total					**3.781**	**Total**					**119.713**	**Total**					**2.014**
Average Power Consumption In Active Mode (W)					13.583	Average Power Consumption In Standby Mode (W)					5.163	Average Power Consumption In PSM (W)					1.563
												Average Power		5.0703			

I have a raspberry pi 4 that I am not currently using. I could bring that in and do some testing with the light weight OS to see what kind of power it is actually drawing. If it is near that value I found online of 3.0-4.5W, this strategy simply will not work.

Regardless of whether I use the Pi or another MPU, I will likely be needing a separate MCU for analog and PWM. I know how to program these things when on an Arduino and in the Arduino environment. However I do not know real embedded C programming. That would likely be useful to learn for this project.

https://www.electronicshub.org/basics-of-embedded-c-program/

Also, the PI does have 2 PWM channels. This means that instead of using a second MCU to control the ADC and PWM, I only need an external ADC for the PI

I am realizing that the camera I was looking at is a slightly worse resolution than the previously selected Camera. The CMV4000 is 2048x2048 (4.1MP) while the Python 5000 was 5MP

Jetson Nano

The Jetson Nano is a computer similar to a Raspberry Pi. One notable difference is that it has a GPU.
For this reason it is well suited to machine vision applications.

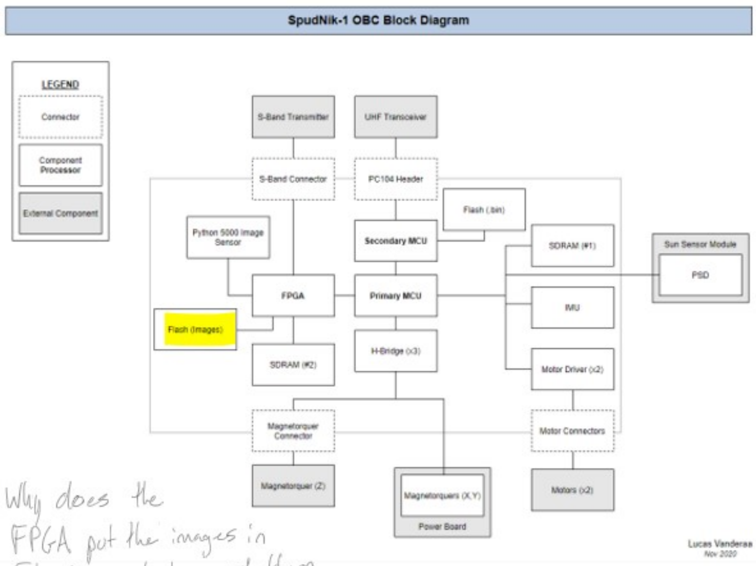
The Nano does have USB 3.0 ports, I2C, SPI, and UART

The power requirements listed are 5-10W

Overall the Jetson is more powerful than a PI, however it seems like it would be overkill for this application. While this project does include a vision system, it is simply taking the images and transmitting them. This does not require a full GPU.

I want to better understand the previous FPGA system to send the image to the S-Band transmitter. I so not understand the output of the Python image sensor, what the FPGA does, and how it interacts with the MCU

Type	Function Name	Description
Interrupt Functions	*bool powerCheck()*	• Checks only the power level, to be performed before switching modes. • Returns True/False
	int modeSelect ()	• Uses variables received from subsystems and truth table to determine what mode the CubeSat should be in. • Will implement protocols depending on which mode is selected. • Returns an integer value representative of the mode to be used as arguments in other functions.
	float sensorData ()	• Checks all sensor data and converts to float values for storage. • This function will be overloaded with an *int mode* argument which only checks specific sensors that need to be read.
	void takeImage ()	• Contains the commands for the payload system to take an image. • An argument of *int numPics* will be included to specify how many pictures will be taken during the pass.
	void storePics ()	• Receives image data from FPGA and stores the data in Flash.
	void storeData ()	• Takes all current telemetry data stored in variables, converts to a floating-point value and stores in Flash. • This function will be overloaded with an *int dataType* argument which allows variables to be stored as types other than floating point.
	void storeError ()	• Records the time an error that occurred in memory.
	int scheduler ()	• Checks uplinked schedule, implements new schedule in RTOS.
Idle Functions	*bool systemsCheck ()*	• Checks communication with all systems periodically. • Returns True/False • This function will be overloaded with an *int mode* argument to check communications between specific subsystems based on what is active.
	bool checkCRC ()	• Periodically checks CRC value of current software to check for corruption.



Why does the FPGA put the images in Flash just to read them again to send to S-band

Figure 8.8: OBC Block Diagram

I cant find any documentation of what the FPGA does and why it is required. If I could understand what it is actually doing I may be able to look for alternative simpler methods

NOIP1SN5000A

Data Order for P1–SN/SE/FN, P3–SN/SE

To read out the image data through the output channels, the pixel array is organized in kernels. The kernel size is sixteen pixels in x–direction by one pixel in y–direction. The data order in 8–bit mode is identical to the 10–bit mode.

Figure 33 indicates how the kernels are organized. The first kernel (kernel [0, 0]) is located in the bottom left corner (front view on top of the package). The data order of this image data on the data output channels depends on the subsampling mode.

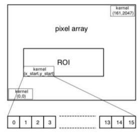


Figure 33. Kernel Organization in Pixel Array – Top View

- P1–SN/SE/FN, P3–SN/SE: Subsampling Disabled
  - 8 LVDS Output Channels (P1–SN/SE/FN only)

The image data is read out in kernels of 16 pixels in x–direction by one pixel in y–direction. One data channel output delivers two pixel values of one kernel sequentially.

Figure 34 shows how a kernel is read out over the eight output channels. For even positioned kernels, the kernels are read out ascending, while for odd positioned kernels the data order is reversed (descending).

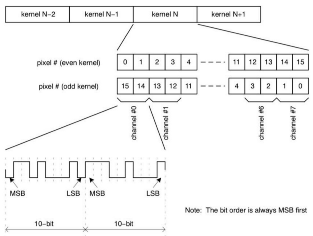


Figure 34. P1–SN/SE/FN: 8 LVDS Data Output Order when Subsampling is Disabled

www.onsemi.com
41

I skimmed through the data sheet of the Python Camera to try to get an understanding of what its output is and then what the FPGA is for.

I found the specifications for the output of the pixel values as seen above.

The FPGA could be responsible for organizing the 8 channel outputs into a single stream of bits.

This way the data could be sent to the S-Band transmitter to be sent to the ground. That would be the main reason I could see. I'm sure there are other formatting things for the output of the camera that would need to be considered, however I don't see why this could not be done in post process of the ground once the image is received. To me it makes more sense to just send raw output data to the S-Band transmitter which can be downloaded and processed afterwards

Technically, the camera can be configured to only use one channel. In this situation, my assumed use of the FPGA would be unnecessary.

NOIP1SN5000A

- 1 LVDS Output Channel

Figure 37 shows how a kernel is read out over 1 output channel. Eight adjacent channels are multiplexed into one channel. For even positioned kernels, the kernels are read out ascending but in sets of 8 even and 8 odd pixels, while for odd positioned kernels the data order is reversed (descending – in sets of 8 odd and 8 even pixels).

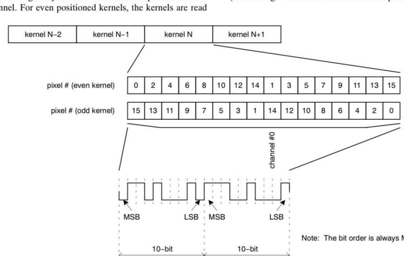
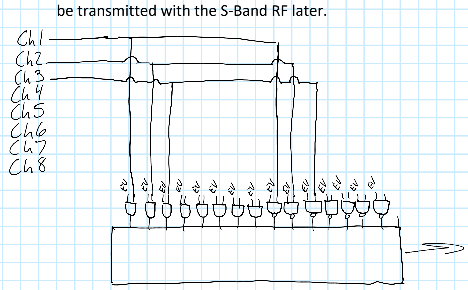
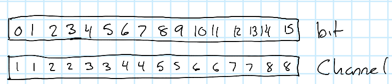


Figure 37. P1–SN/SE/FN, P3–SN/SE: 1 LVDS Data Output Order when Subsampling is Disabled

Presumably in this configuration, the image could be sent directly to the S-Band transmitter.

I guess another aspect to this is time. The satellite is only over PEI for 4min in a 90min orbit. The FPGA may be to convert the 8 channels to 1 and store as many images in memory as possible so that they can be transmitted with the S-Band RF later.





This would be slow
Plus how would you ensure
the clock is X times
faster than the Cam clock

General Take Always

From my research so far, I have found that:

- It is very unlikely to find a MPU with the required ADC channels build in. The largest I have found is a MPU chip (Not dev board) that has 8 ADC channels available. For this reason, it is expected that an external ADC or MCU will be required
- MPU that accommodate USB3.0 are also limited. I have found that some Raspberry Pi versions have USB3.0.
- MPU are going to have higher power requirements than the current MCU system. From what I have found in a power usage spreadsheet is that there is a relatively small tolerance between the required power and the supplied power. I also do not know how to accurately determine the power of the MPU without testing, as I am sure that the OS and tasks will affect the power drawn
- I have found that a Raspberry Pi 4 would have the required interfaces to be suitable for this project, with an external ADC. The main issue with that plan is the power usage.