The communication between the ground and satellite is separated into packets. This document specifies the form and function of these packets.

Packet Protocol

Detailed Description of Communication Packets

Revision: 1.0.2

Bradley Davis

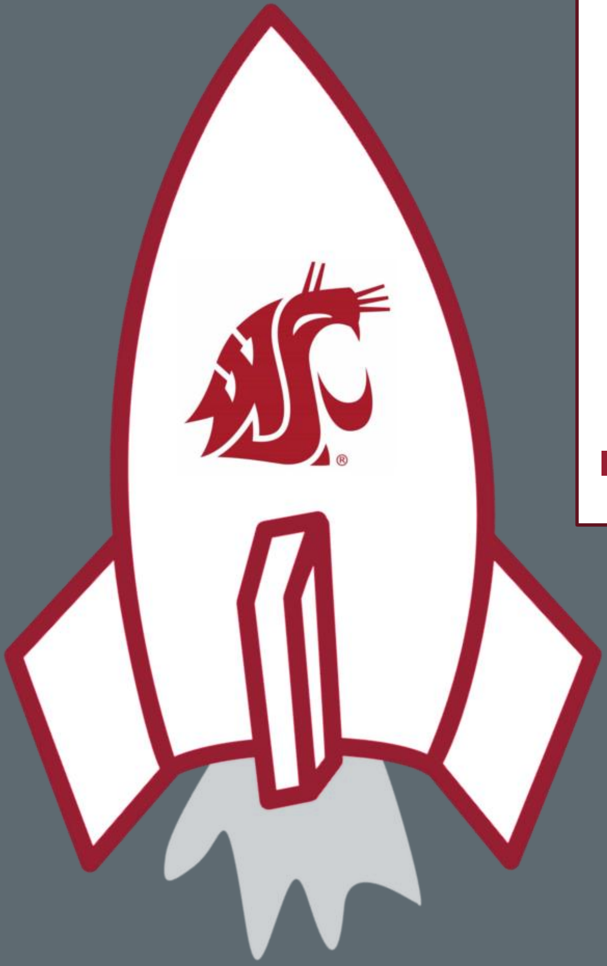

Table of Contents

1 Packet Header

The function of the packet header is to convey the recipient and sender of each packet. It also indicates whether the packet is part of a larger packet (multipacket or file transfer). Throughout this document, multipacket represents a set of two or more packets primarily used to transfer files.

The header is two bytes long. The first byte contains the recipient, sender, and multipacket flag, see bit allocation in table 1. The second byte is the length of the body in number of dwords (32b). The choice of dwords allows future expandability of packet lengths up to $2^8 \cdot 32b = 1024B$, the current RCS limits packets to a length of 253B. A packet might require padding at the end of the body to increase the body length to a multiple of single dwords.

Table 1: Packet Header First Byte

Bit index	7	6	5	4	3	2	1	0
Function	Sender `0x0`: Ground `0x1`: ADCS `0x2`: IFJR `0x3`: IHU `0x4`: PMIC `0x5`: RCS			Recipient `0x0`: Ground `0x1`: ADCS `0x2`: IFJR `0x3`: IHU `0x4`: PMIC `0x5`: RCS				Multipacket Flag

1.1 Multipacket Additional Header

For a multipacket, there are an additional two bytes appended to the packet header. These represent a 16b long serial number for the packet. By analyzing the serial numbers of all packets received for a multipacket, the recipient can figure which packets were lost and request those specific packets from the sender.

2 Packet Body

The body of a packet is unique to every system and command. The details of every command are explained below.

2.1 Multipacket

The function of a multipacket is to transfer data that is longer than a single packet can support. The primary uses of a multipacket are images and processor binaries.

Every packet, except the last packet, will be completely full. There is a low chance that the last packet is also completely full. From the file size given in the first packet and the knowledge on the length of packets, the recipient can immediately figure the number of packets it is going to receive and how full the last packet is going to be.

2.1.1 First Packet

The first packet provides details on the file being transferred, including length and file name. The first 4 bytes represents the file length as an unsigned integer, file sizes are limited to $2^{32}b \approx 4GB$. The next several bytes, until a null character is reached, represents the filename as a string. Immediately following the null character is the start of the file.

2.1.2 Second to (n – 1)th Packet

The second and onward packet, to one before the last, just contains the data of the file.

2.1.2 nth Packet

The last packet also contains just the data of the file; however, there is a likely possibility that this packet will not be completely full. If so, the same padding rules apply as normal packets: add padding until the body is an even multiple of dwords. Once this packet is received, or communication times out, the recipient will identify which packets were lost in transmission and request them again using their serial numbers

2.1.3 Lost Packet Request

If a packet is lost, the recipient will send a request for retransmission of that specific packet. The body of this message will be a byte to indicate lost packets (`0xFF`) and the serial numbers of lost packets sequentially appended. If number of lost packets exceeds the number that fit in a single packet, change the lost packet byte to (`0xFE`) and send additional requests. Once all packets are accounted for, send a multipacket transfer success message and began assembling packets.

2.1.4 Multipacket Transfer Success

Upon successful completion of a multipacket transfer, send a message to the sender to inform of the success so it can resume normal operation, or possibly begin another multipacket. The body of this message is (`0x00 0x01 0xF6 0x80`), which is the Unicode character for 🚀.

2.2 ADCS Commands

Each command has an uplink and downlink form. Uplink is the ground to the ADCS. Downlink is the ADCS to the ground. The commands are used to get health data, get diagnostic data, get raw attitude sensor values, and request a maneuver.

2.2.1 Diagnostic Report

A full diagnostic report is requested from the ADCS with the command ID `0x00`, see table 2. The ADCS replies with command ID `0x80`, current position, attitude, coil driver PWM count, and coil currents as listed in table 3. The coil PWM count in the raw value at which the PWM output is set low.

Table 2: ADCS Diagnostic Report Uplink

Index	0x00
Function	0x00

Table 3: ADCS Diagnostic Report Downlink

Index	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08
Function	0x80	Lat (int32, (10^{-4}) min/LSB)				Long (int32, (10^{-4}) min/LSB)			

Index	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F	0x10	0x11
Function	Roll (uint24, 2^{24} = 2π)			Pitch (uint24, 2^{24} = 2π)			Yaw (uint24, 2^{24} = 2π)		

Index	0x12	0x13	0x14	0x15	0x16	0x17	0x18	0x19	0x1A
Function	X Count (int16)		Y Count (int16)		Z Count (int16)		X Current (int24, 1nA/LSB)		

Index	0x1B	0x1C	0x1D	0x1E	0x1F	0x20
Function	Y Current (int24, 1nA/LSB)			Z Current (int24, 1nA/LSB)		

2.2.2 IMU Sensor

The IMU sensor data is requested from the ADCS with the command ID `0x01` and sensor ID, see table 4. The ADCS replies with command ID `0x81`, and raw sensor values as listed in table 5.

Table 4: ADCS IMU Sensor Uplink

Index	0x00	0x01
Function	0x01	Sensor ID 0x0: IMU 0 0x1: IMU 1 0x2: IMU 2 0x3: IMU 3

Table 5: ADCS IMU Sensor Downlink

Index	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08
Function	0x81	X Mag (raw)		Y Mag (raw)		Z Mag (raw)		X Gyro (raw)	

Index	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F	0x10
Function	Y Gyro (raw)		Z Gyro (raw)		X Accel (raw)		Y Accel (raw)	

Index	0x11	0x12
Function	Z Accel (raw)	

2.2.3 Maneuver to a Point

A maneuver to a point is requested from the ADCS with the command ID `0x02`, reference frame, and position see table 6. The ADCS attempts to point the satellite at the fixed point. The ADCS replies with command ID `0x82`, and estimated time of completion as listed in table 7. If multiple maneuvers are requested, the most recently received will be performed.

Table 6: ADCS Maneuver to a Point Uplink

Index	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07	0x08	0x09
Function	0x02	Reference Frame 0x0: Earth 0x1: Space	Latitude (int32, (10^{-4}) min/LSB)				Longitude (int32, (10^{-4}) min/LSB)			

Table 7: ADCS Maneuver to a Point Downlink

Index	0x00	0x01	0x02	0x03	0x04
Function	0x82	ETC (uint32, Unix Time)			

2.2.4 Maneuver to an Attitude

A maneuver to an attitude is requested from the ADCS with the command ID `0x03`, reference frame, and position see table 8. The ADCS attempts to point the satellite at the fixed attitude. The ADCS replies with command ID `0x83`, and estimated time of completion as listed in table 9. If multiple maneuvers are requested, the most recently received will be performed.

Table 8: ADCS Maneuver to an Attitude Uplink

Index	0x00	0x02	0x03	0x04	0x05	0x06	0x07
Function	0x03	Roll (uint24, $2^{24} = 2\pi$)			Pitch (uint24, $2^{24} = 2\pi$)		

Index	0x08	0x09	0x0A
Function	Yaw (uint24, $2^{24} = 2\pi$)		

Table 9: ADCS Maneuver to an Attitude Downlink

Index	0x00	0x01	0x02	0x03	0x04
Function	0x83	ETC (uint32, Unix Time)			

2.3 IFJR Commands

Each command has an uplink and downlink form. Uplink is the ground to the IFJR. Downlink is the IFJR to the ground. The commands are used to get health data, get diagnostic data, and request a reprogram.

2.3.1 Status Report

A simple status report is requested from the IFJR with the command ID `0x00`, see table 10. The IFJR replies with command ID `0x80`, current code version numbers, and SD card used size as listed in table 11.

Table 10: IFJR Status Uplink

Index	0x00
Function	0x00

Table 11: IFJR Status Downlink

Index	0x00	0x01	0x02	0x03	0x04
Function	0x80	ADCS	IHU	PMIC	RCS

Index	0x05	0x06	0x07	0x08	0x09
Function	SD Card Used Size (uint40, 1B/LSB)				

2.3.2 Diagnostic Report

A full diagnostic report is requested from the IFJR with the command ID `0x01` and device ID, see table 12. The IFJR replies with command ID `0x81`, and current code version numbers as listed in table 13.

Table 12: IFJR Diagnostic Report Uplink

Index	0x00	0x01
Function	0x01	Device ID 0x0: ADCS 0x1: IFJR 0x2: IHU 0x3: PMIC 0x4: RCS

Table 13: IFJR Diagnostic Report Downlink

Index	0x00	0x01	0x02	0x03	0x04	0x05
Function	0x81	Version	Last Program (uint32, Unix Time)			

2.3.3 Reprogram Device

A device reprogram is requested from the IFJR with the command ID `0x02` and device ID, see table 14. The IFJR replies with command ID `0x82`, and programming duration as listed in table 15.

Table 14: IFJR Reprogram Device Uplink

Index	0x00	0x01
Function	0x02	Device ID 0x0: ADCS 0x1: IFJR 0x2: IHU 0x3: PMIC 0x4: RCS

Table 15: IFJR Reprogram Device Downlink

Index	0x00	0x01	0x02	0x03
Function	0x82	Time (uint24, 1ms/LSB)		

2.4 IHU Commands

Each command has an uplink and downlink form. Uplink is the ground to the IHU. Downlink is the IHU to the ground. The commands are used to get health data, get diagnostic data, and standard telemetry. The standard telemetry is usually not needed to be requested as it is sent out at a regular interval.

2.4.1 Diagnostic Report

A full diagnostic report is requested from the IHU with the command ID `0x00`, see table 16. The IHU replies with command ID `0x80` and current operation mode, current time, reset count, and SD card used size as listed in table 17.

Table 16: IHU Diagnostic Report Uplink

Index	0x00
Function	0x00

Table 17: IHU Diagnostic Report Downlink

Index	0x00	0x01	0x02	0x03	0x04	0x05	0x06
Function	0x80	Mode	Current Time (uint32, Unix Time)				Reset Count

Index	0x07	0x08	0x09	0x0A	0x0B
Function	SD Card Used Size (uint40, 1B/LSB)				

2.4.2 Standard Telemetry

A standard telemetry packet is requested from the IHU with the command ID `0x01`, see table 18. The IHU replies with command ID `0x81`, current operation mode, and telemetry from each subsystem as listed in table 19. A standard telemetry packet is sent at regular intervals during the event ihuPeriodic. Normally, a request for a standard telemetry packet does not need to be requested. Temperature readings are signed 8b integers with 1°C/LSB. Voltages and currents are each unsigned 12b integers, forming 3 bytes together. Voltages are 2mV/LSB and currents are 1mA/LSB. Powers (including heater output) are unsigned 8b integers with 50mW/LSB.

Table 18: IHU Standard Telemetry Uplink

Index	0x00
Function	0x01

Table 19: IHU Standard Telemetry Downlink

Index	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
Function	0x81	Mode	Current Time (uint32, Unix Time)				Reset Count	IHU Temp

Index	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
Function	IHU SD Card Used Size (uint40, 1B/LSB)					Error Status	ADCS Temp	ADCS Status

Index	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
Function	Lat (int32, (10^{-4}) min/LSB)				Long (int32, (10^{-4}) min/LSB)			

Index	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
Function	Roll (uint24, $2^{24} = 2\pi$)			Pitch (uint24, $2^{24} = 2\pi$)			Yaw (uint24, $2^{24} = 2\pi$)	

Index	0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
Function	Yaw	IFJR Temp	PMIC Temp	Battery 0 Voltage & Current			Battery 1 V & I	

Index	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
Function	Batt 1 V & I	Batt 0 Temp	Batt 1 Temp	Batt 0 Heat	Batt 1 Heat	Solar Panel 0 Voltage & Current		

Index	0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
Function	Solar Panel 1 Voltage & Current			Solar Panel 2 Voltage & Current			Solar Panel 3 V & I	

Index	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
Function	PV 3 V & I	Pin (uint16) (1mW/LSB)		Pout (uint16) (1mW/LSB)		3.3V Rail Voltage & Current		

Index	0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47
Function	5.0V Rail Voltage & Current			PMIC Channels' State		RCS Temp	RCS Status	TX Power

Index	0x48	0x49	0x4A	0x4B
Function	RX Power	RX SNR	Payload Frames in TX Queue	

2.5 EPS Commands

Each command has an uplink and downlink form. Uplink is the ground to the EPS. Downlink is the EPS to ground. The commands are used to get diagnostic data from the EPS.

2.5.1 Diagnostic Report

A full diagnostic report is requested from the EPS with the command ID `0x00`, see table 20. The EPS replies with command ID `0x80`, state of primary and secondary PMIC, battery voltages, currents, and temperatures, solar panel voltages, currents, and temperatures, voltage rail voltages and currents, regulator temperatures, power channel currents and switch status, and net energy for the day as listed in table 21. The net energy is reset at 0:00:00 UTC every day.

Voltages are unsigned 16b integers with 100μV/LSB. Currents are unsigned 16b 100μA/LSB, unless otherwise defined. Temperatures are signed 8b integers with 1°C/LSB. Power channel switch status is a unsigned 16b integer, each bit corresponds to a single channel: [15..10] 3V-0..5, [9..8] 5V-0..1, [7..2] U-0..5, [1..0] Heater P..S. "P" and "S" are used to indicate reference to the primary or secondary PMIC, hardware defined as their software roles might switch.

Table 20: EPS Diagnostic Report Uplink

Index	0x00
Function	0x00

Table 21: EPS Diagnostic Report Downlink

Index	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
Function	0x80	1st/2nd Role	Battery P Voltage		Battery P Current (int16)		Battery S Voltage	

Index	0x08	0x09	0x0A	0x0B	0x0C	0x0D	0x0E	0x0F
Function	Battery S Current		Batt P Temp	Batt S Temp	Solar Panel 0 Voltage		Solar Panel 0 Current	

Index	0x10	0x11	0x12	0x13	0x14	0x15	0x16	0x17
Function	Solar Panel 1 Voltage		Solar Panel 1 Current		Solar Panel 2 Voltage		Solar Panel 2 Current	

Index	0x18	0x19	0x1A	0x1B	0x1C	0x1D	0x1E	0x1F
Function	Solar Panel 3 Voltage		Solar Panel 3 Current		PV 0 Temp 0	PV 0 Temp 1	PV 0 Temp 2	PV 1 Temp 0

Index	0x20	0x21	0x22	0x23	0x24	0x25	0x26	0x27
Function	PV 1 Temp 1	PV 1 Temp 2	PV 2 Temp 0	PV 2 Temp 1	PV 2 Temp 2	PV 3 Temp 0	PV 3 Temp 1	PV Temp 2

Index	0x28	0x29	0x2A	0x2B	0x2C	0x2D	0x2E	0x2F
Function	3V Rail P Voltage		3V Rail P Current		3V Rail S Voltage		3V Rail S Current	

Index	0x30	0x31	0x32	0x33	0x34	0x35	0x36	0x37
Function	5V Rail P Voltage		5V Rail P Current		5V Rail S Voltage		5V Rail S Current	

Index	0x38	0x39	0x3A	0x3B	0x3C	0x3D	0x3E	0x3F
Function	3V Reg P Temp	3V Reg S Temp	5V Reg P Temp	5V Reg S Temp	MPPT 0 P Temp	MPPT 0 S Temp	MPPT 1 P Temp	MPPT 1 S Temp

Index	0x40	0x41	0x42	0x43	0x44	0x45	0x46	0x47
Function	3V-0 P Current		3V-0 S Current		3V-1 P Current		3V-1 S Current	

Index	0x48	0x49	0x4A	0x4B	0x4C	0x4D	0x4E	0x4F
Function	3V-2 P Current		3V-2 S Current		3V-3 P Current		3V-3 S Current	

Index	0x50	0x51	0x52	0x53	0x54	0x55	0x56	0x57
Function	3V-4 P Current		3V-4 S Current		3V-5 P Current		3V-5 S Current	

Index	0x58	0x59	0x5A	0x5B	0x5C	0x5D	0x5E	0x5F
Function	5V-0 P Current		5V-0 S Current		5V-1 P Current		5V-1 S Current	

Index	0x60	0x61	0x62	0x63	0x64	0x65	0x66	0x67
Function	U-0 P Current		U-0 S Current		U-1 P Current		U-1 S Current	

Index	0x68	0x69	0x6A	0x6B	0x6C	0x6D	0x6E	0x6F
Function	U-2 P Current		U-2 S Current		U-3 P Current		U-3 S Current	

Index	0x70	0x71	0x72	0x73	0x74	0x75	0x76	0x77
Function	U-4 P Current		U-4 S Current		U-5 P Current		U-5 S Current	

Index	0x78	0x79	0x7A	0x7B	0x7C	0x7D	0x7E	0x7F
Function	Battery Heater P Current		Battery Heater S Current		Channel Switch Status		Energy (int16) (10J/LSB)	

2.6 Payload Commands

Each command has an uplink and downlink form. Uplink is the ground to the payload. Downlink is the payload to the ground. The commands are used to get diagnostic data, take and download photos, and download experimental data. To change parameters of the experiment, such as measurement interval, send a file up that the experiment will understand.

2.6.1 Diagnostic Report

A full diagnostic report is requested from the payload with the command ID 0x00, see table 22. The payload replies with command ID 0x80 as listed in table 23.

Table 22: Payload Diagnostic Report Uplink

Index	0x00
Function	0x00

Table 23: Payload Diagnostic Report Downlink

Index	0x00
Function	0x80

2.6.2 Take Photo

A take photo event is requested from the payload with the command ID `0x01`, desired time to expose, and camera ID, see table 24. The payload replies with command ID `0x81`, status, and photo ID as listed in table 25. Multiple request can be stacked, and they will all be taken. To retrieve a photo, use the download photo command.

Table 24: Payload Take Photo Uplink

Index	0x00	0x01	0x02	0x03	0x04	0x05
Function	0x01	Time to Expose (uint32, Unix Time)				Camera ID 0x0: Ground Wide 0x1: Germination

Table 25: Payload Take Photo Downlink

Index	0x00	0x01	0x02	0x03
Function	0x81	Status	Photo ID	

2.6.3 Download Photo

A photo download is requested from the payload with the command ID `0x02` and photo ID, see table 26. The payload replies with a multipacket file transfer, see 2.1. If a photo ID of $2^{16} - 1$ is requested, the next photo not yet transferred will be sent. The specifics of the file being transferred is explained in the Image Format document.

Table 26: Payload Download Photo Uplink

Index	0x00	0x01	0x02
Function	0x02	Photo ID	

2.6.4 Experiment Report

An experiment report is requested from the payload with the command ID `0x03` and photo ID, see table 27. The payload replies with a multipacket file transfer, see 2.1. If an experiment ID of $2^8 - 1$ is requested, the next experiment report not yet transferred will be sent. The specifics of the file being transferred is explained in each experiment's documentation.

Table 27: Payload Experiment Report Uplink

Index	0x00	0x01
Function	0x03	Experiment ID 0x0: Radiation 0x1: Temperature 0x2: Germination

2.7 RCS Commands

Each command has an uplink and downlink form. Uplink is ground to the RCS. Downlink is RCS to ground. The commands are used to get diagnostic data and change parameters of the radios.

2.7.1 Diagnostic Report

A full diagnostic report is requested from the RCS with the command ID `0x00`, see table 28. The RCS replies with command ID `0x80`, transmitting radio power, receiving radio power, radio center frequencies, daily bad packet count, and receiver signal to noise ratio as listed in table 29. The daily bad packet count resets at 0:00:00 UTC every day. The bad packet count includes partial packets received, and packets with wrong cyclic redundancy check. Power data are unsigned 16b integers with 1mW/LSB. SNR is an unsigned 8b integer with 0.1dB/LSB

Table 28: RCS Diagnostic Report Uplink

Index	0x00
Function	0x00

Table 29: RCS Diagnostic Report Downlink

Index	0x00	0x01	0x02	0x03	0x04	0x05	0x06	0x07
Function	0x80	TX Power		RX Power		TX Center Frequency (uint24, 100Hz/LSB)		

Index	0x08	0x09	0x0A	0x0B	0x0C	0x0D
Function	RX Center Frequency (uint24, 100Hz/LSB)			Bad Packet Count (uint16)		RX SNR

2.7.2 Change Radio

A change to a radio is requested from the RCS with the command ID `0x01`, radio ID, max transmit power, and center frequency, see table 30. The RCS replies with command ID `0x81` and status as listed in table 31.

Table 30: RCS Change Radio Uplink

Index	0x00	0x01	0x02	0x03	0x04	0x05	0x06
Function	0x01	ID	Power (uint16) (1mW/LSB)		Center Frequency (uint24, 100Hz/LSB)		

Table 31: RCS Change Radio Downlink

Index	0x00	0x01
Function	0x81	Status