Hybrid Control System Packet Format

Matteo Golin
`matteo.golin@gmail.com`

Created: September 8th, 2024
Modified: June 20, 2025

1 Introduction

The hybrid control system used by CU InSpace requires network communication between nodes. The nodes consist of a control input box at the ground control station, the pad control system at the rocket's launch pad and a user interface that can be placed anywhere with connection to the network.

In this set up, the pad control box is considered a server. It provides sensor measurements as well as information about the control system's state. It is the only source of truth in the system. No other node should be responsible for storing state information about the control system.

The UI and input control box form clients. This creates a distinction between two types of client nodes:

1. A control node, which is able to issue commands to the control system. There should only be one control node in the system.
2. A telemetry node, which consumes telemetry data for either logging or display. There can be as many of these nodes in the system as necessary.

These two distinct types of client nodes inform the decision to distinguish two primary message categories: Control (Section 3) and Telemetry 4.

1.1 Control

Control messages provide a specification for issuing commands to the control system and receiving a response. These commands are for actuating valves and servos, as well as for progressing the arming level of the control system.

Control messages issued from the client to the server are replied to with an acknowledgement containing the status of the request.

1.2 Telemetry

Telemetry messages provide a specification for sending information about the control system state. This includes sensor measurements, the current arming level, actuator states and warning messages.

Telemetry messages are not requested, unlike the control commands. They are sent in a continuous stream from the server to the client.

1.3 Notices

It should be noted that this specification is designed under the assumption that all messages are sent over a local network. Telemetry messages are intended to be sent over UDP, while control messages are intended to be sent over TCP.

All fields are to be assumed as having little-endian byte ordering unless specified otherwise.

2 Headers

In order for the recipient to reliably parse incoming messages, a standardized header will precede all communication messages. This header will specify the type of message which follows, giving the recipient enough information to parse it. This structure will look like Figure 1.

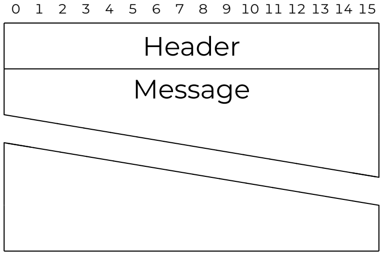

Figure 1: Packet structure

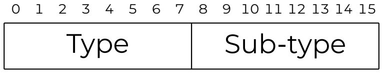

Figure 2: Message header

Type: The type of message being sent. See Table 1 for valid types.

Sub-type: The sub type of message being sent. See Table 1 for valid sub-types associated with each type.

Type	Value	Sub-Type	Value	Description
CONTROL	0	ACT_REQ	0	See section 3.1.1
		ACT_ACK	1	See section 3.1.2
		ARM_REQ	2	See section 3.2.1
		ARM_ACK	3	See section 3.2.2
TELEMETRY	1	TEMPERATURE	0	See section 4.1.1
		PRESSURE	1	See section 4.1.2
		MASS	2	See section 4.1.3
		THRUST	3	See section 4.1.4
		ARMING_STATE	4	See section 4.2
		ACT_STATE	5	See section 4.3
		WARNING	6	See section 4.4
		CONTINUITY	7	See section 4.5
		CONN_STATUS	8	See section 4.6

Table 1: Valid types and sub-types

.

3 Control

Control messages are intended to be sent using the TCP protocol. This is because control commands must be guaranteed to be delivered since they control safety critical components of the control system. The control client and pad server must also be able to reliably tell when the connection used for communication has been severed. TCP provides such reliability.

The port used for control messages is `50001`.

3.1 Actuation

The pad control system is composed of several actuators which need to be controlled from the control input box at the ground station.

All of the actuators on the system are binary actuators with only an "on" or "off" state.

Actuators will be assigned a unique numerical ID agreed upon by all nodes on the network.

All actuation requests must be responded to with an actuation acknowledgement indicating the status of the command.

Name	Value	Description
ACT_OK	0	The pad control system has put the actuator in the requested state.
ACT_DENIED	1	The current arming level is too low to operate the request actuator.
ACT_DNE	2	The actuator ID is not associated with an actuator.
ACT_INV	3	An invalid state was requested.

Table 2: Valid actuation acknowledgement statuses

3.1.1 Actuation Request

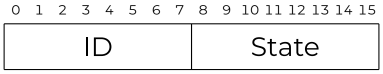

Figure 3: Actuation request format

ID: The unique numerical identifier associated with the actuator who this request is destined for.

State: The state to put the actuator in. Valid states are listed in Table 5.

3.1.2 Actuation Acknowledgement

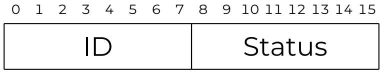

Figure 4: Actuation acknowledgement format

ID: The unique numerical identifier associated with the actuator who this acknowledgement is from.

Status: The status of the request that is being acknowledged. Valid status are listed in Table 2.

3.2 Arming

Arming messages are used to advance the current level of arming in the pad control system. Many arming sequences must take place remotely from the control input box at the ground station, hence their inclusion in the specification.

There are two message types for arming: the arming request and the arming acknowledgement. All arming requests are to be responded to with an arming acknowledgement indicating the status of the request.

Name	Value	Description
ARM_OK	0	The arming request was processed and the pad control system has changed arming level.
ARM_DENIED	1	The arming request could not be completed because the current arming level is not high enough for that level progression, or the progression must be caused via an actuator command.
ARM_INV	2	Invalid arming level was specified.

Table 3: Valid arming acknowledgement statuses

3.2.1 Arming Request

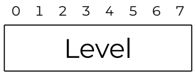

Figure 5: Arming request format

Level: The arming level that the client is requesting be moved to. A list of valid arming levels can be found in Table 4. Only the `ARMED_VALVES` and the `ARMED_IGNITION` arming levels can be requested with this message type. The remaining arming levels are obtained by specific actuator interactions.

3.2.2 Arming Acknowledgement

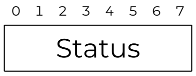

Figure 6: Arming acknowledgement format

Status: The status of the arming request which this message is acknowledging. A list of valid statuses can be found in Table 3.

4 Telemetry

All telemetry messages are designed to be sent over UDP using multicast functionality. This will allow the server implementation to send a single UDP message which can be delivered to all subscribers.

In addition, the use of the UDP protocol is well-suited for telemetry applications. Measurements are made and transmitted multiple times in a tenth of a second. It is not worth the overhead of TCP to have acknowledgements for each of these messages, since if one is lost many will follow and the information will become obsolete quickly.

The multicast address for telemetry is `224.0.0.10`, using port `50002`.

4.1 Measurements

The hybrid control system measures four key pieces of information:

1. Temperature
2. Pressure
3. Mass
4. Thrust

These measurements give the operator insight into whether or not the system is becoming unstable, if filling is proceeding nominally, etc.

All measurements are only temporally valid, so they must be associated with a time stamp indicating the time of measurement.

4.1.1 Temperature

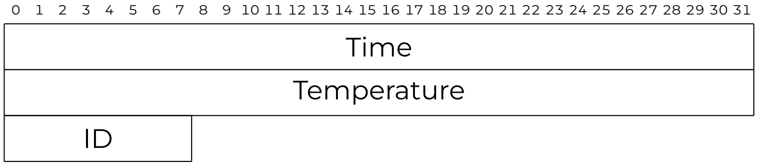

Figure 7: Temperature measurement format

Time: A time stamp measured in milliseconds since the system received power. This field is an unsigned 32-bit integer.

Temperature: Temperature measured in millidegrees Celsius. This field is a signed 32-bit integer.

ID: The numerical ID of the sensor which took the measurement. This field is an unsigned 8-bit integer.

4.1.2 Pressure

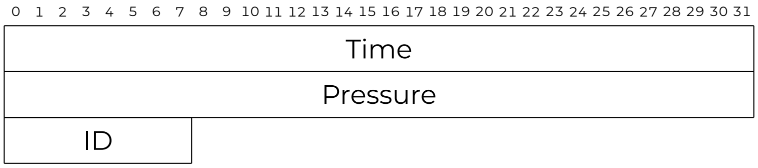

Figure 8: Pressure measurement format

Time: A time stamp measured in milliseconds since the system received power. This field is an unsigned 32-bit integer.

Pressure Pressure measured in thousandths of PSI (milli-PSI). This field is a signed 32-bit integer.

ID: The numerical ID of the sensor which took the measurement. This field is an unsigned 8-bit integer.

4.1.3 Mass

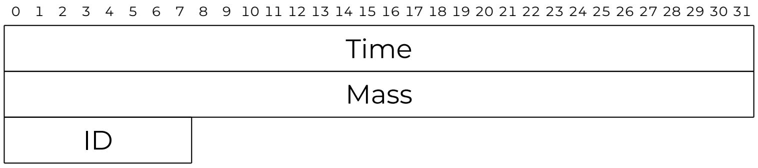

Figure 9: Mass measurement format

Time: A time stamp measured in milliseconds since the system received power. This field is an unsigned 32-bit integer.

Mass Mass measured in grams. This field is a signed 32-bit integer.

ID: The numerical ID of the sensor which took the measurement. This field is an unsigned 8-bit integer.

4.1.4 Thrust

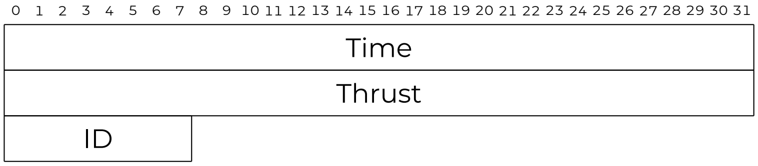

Figure 10: Thrust measurement format

Time: A time stamp measured in milliseconds since the system received power. This field is an unsigned 32-bit integer.

Thrust Engine thrust measured in Newtons. This field is an unsigned 32-bit integer.

ID: The numerical ID of the sensor which took the measurement. This field is an unsigned 8-bit integer.

4.2 Arming State

The hybrid control system's arming state should be known at all times over telemetry.

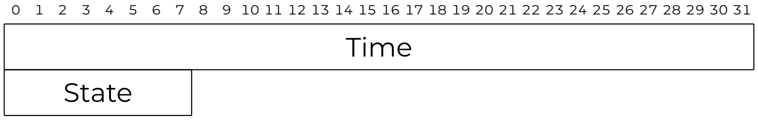

Figure 11: Arming state message format

Time: A time stamp measured in milliseconds since the system received power. This field is an unsigned 32-bit integer.

State: The arming state of the control system.

State	Value
ARMED_PAD	0
ARMED_VALVES	1
ARMED_IGNITION	2
ARMED_DISCONNECTED	3
ARMED_LAUNCH	4

Table 4: Valid arming states

4.3 Actuator State

The state of the hybrid control system's actuators should be known at all times over telemetry.

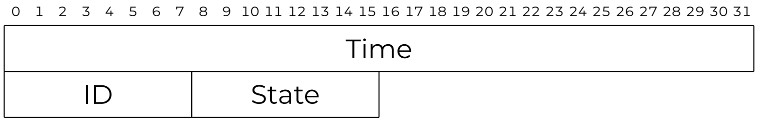

Figure 12: Actuator state message format

Time: A time stamp measured in milliseconds since the system received power. This field is an unsigned 32-bit integer.

ID: The unique numerical ID of the actuator whose state is being sent in this message.

State: The state of the actuator. See Table 5 for the valid actuator states.

State	Value	Description
OFF	0	The actuator is off, or close.
ON	1	The actuator is on, or open.

Table 5: Valid actuator states

4.4 Warnings

Warnings are issued when measurements exceed a certain threshold.

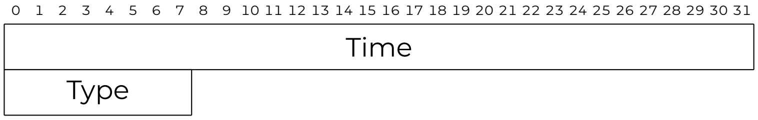

Figure 13: Warning message format

Time: A time stamp measured in milliseconds since the system received power. This field is an unsigned 32-bit integer.

Type: The warning type. A list of valid types can be found in Table 7.

Actuator Name	ID	Description
Igniter	0	The igniter that starts the flame for launch
XV-1	1	Solenoid valve
XV-2	2	Solenoid valve
XV-3	3	Solenoid valve
XV-4	4	Solenoid valve
XV-5	5	Solenoid valve (used as fire valve)
XV-6	6	Solenoid valve
XV-7	7	Solenoid valve
XV-8	8	Solenoid valve
XV-9	9	Solenoid valve
XV-10	10	Solenoid valve
XV-11	11	Solenoid valve
XV-12	12	Solenoid valve
Quick disconnect	13	The quick disconnect to separate ground system plumbing from the rocket.
Dump valve	14	The dump valve that safely dumps all the oxidizer

Table 6: Agreed actuator IDs

Name	**Value**	**Description**
HIGH_PRESSURE	0	Pressure levels have exceeded the warning threshold and manual intervention is required.
HIGH_TEMP	1	Temperature levels have exceeded the warning threshold and manual intervention is required.

Table 7: Warning types

4.5 Continuity state

The continuity sensor measures if the igniter is continuous (closed circuit) or not (open circuit, burnt through/fired).

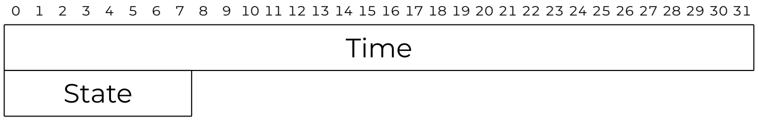

Figure 14: Continuity state message format

Time: A time stamp measured in milliseconds since the system received power. This field is an unsigned 32-bit integer.

State: The state of the continuity sensor. See Table 8 for the valid states.

State	Value	Description
OPEN	0	Continuity sensor is reading low, circuit is open.
CLOSED	1	Continuity sensor is reading high, circuit is closed.

Table 8: Valid continuity sensor states

4.6 Connection Status

Indicates the state of the connection between the control client and the pad server.

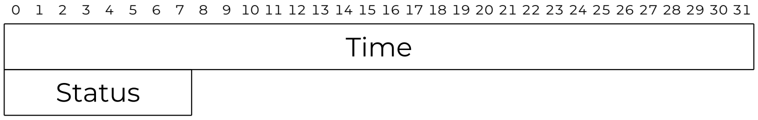

Figure 15: Connection status message format

Time: A time stamp measured in milliseconds since the system received power. This field is an unsigned 32-bit integer.

Status: The status of the connection. See Table 9 for the valid statuses.

Status	Value	Description
CONNECTED	0	The control client is connected through TCP.
RECONNECTING	1	The control client is not connected, but a re-connection attempt is being ma
DISCONNECTED	2	The control client is not connected, re-connection attempts failed.

Table 9: Valid connection statuses

Note that in practice, the `DISCONNECTED` state may never be observed. This is because a failed re-connection attempt can be handled by the pad server as an automatic mission abort trigger, resulting in a complete system shut-down and shut off of all valves.

5 Arming Logic

The control logic that the pad server uses to govern which actuation and arming commands are valid at any given time are based on the arming state in Figure 13.

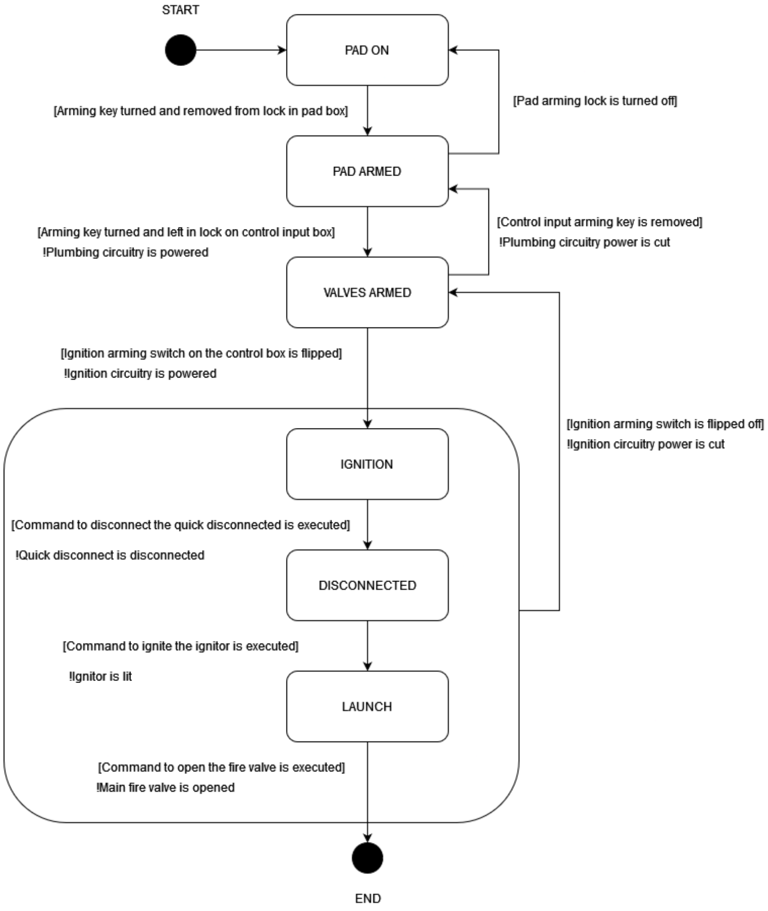

Figure 16: Finite state machine for arming control

State	Available actuator IDs (inclusive)	Description
ARMED_PAD	None	
ARMED_VALVES	1-12	All solenoid valves (XV-1 to XV-12)
ARMED_IGNITION	1-13	All solenoid valves and the quick disconnect
ARMED_DISCONNECTED	1-14	All solenoid valves, quick disconnect and igniter
ARMED_LAUNCH	0-14	All solenoid valves, quick disconnect, igniter and main fire valve

Table 10: Available actuators for each arming state

The quick disconnect is an actuator which upon being switched "on", forcibly disconnects the ground systems plumbing from the oxidizer tank on board the rocket. This operation cannot be reversed.

The igniter is a small piece of solid fuel with nichrome wires running through it. Once it is turned "on", high current is sent through the nichrome wires, igniting a flame underneath the fire valve. This action cannot be reversed.

The main fire valve is a single valve that when opened, allows the oxidizer to flow out of the oxidizer tank on the rocket and into the combustion chamber. It will flow out of the nozzle on the combustion chamber on to the lit igniter, initiating lift-off. This action *obviously* cannot be reversed!