DESIGN STANDARD
FOR

Picatinny Common Lethality Integration Kit (CLIK)

DISTRIBUTION STATEMENT A: Approved for public release: distribution unlimited.

Controlled By: DEVCOM-AC
Distribution Statement: A
POC: Daniel DiMartino,
daniel.f.dimartino.civ@army.mil

DOCUMENT ORIGIN AND APPROVAL RECORD

Prepared by: ________________________________ Date: ______________

<name>, Engineer – DEVCOM-Armaments Center

Prepared by: ________________________________ Date: ______________

<name>, Engineer – DEVCOM-Armaments Center

Reviewed by: _________________________________ Date: ______________

<name>, Engineer – DEVCOM-Armaments Center

Reviewed by: _________________________________ Date: ______________

<name>, Engineer – DEVCOM-Armaments Center

Approved by: _________________________________ Date: ______________

<name>, TBD – Project Manager Close Combat Systems (PM CCS)

REVISION STATUS

Revision	Date	Notes	Approved
X-	2024-10-17	PRE-PRODUCTION RELEASE	
XC-working	2025-04-25	Version 2 Working Copy. Revised from Interface Control Document to Design Standard.	
2.1	2025-09-17	Interim Release: Updates/Refinements based on comments and working group meetings.	
3.0	2026-04-13	Incorporation of Safety Processing for CLIK. In addition, some enhancements/clarifications to mechanical, electrical, and integration levels information. Key updates: • (1.3): Addition of table for Integration Level 1-3 requirements • (2.2): Safety related standards • Section 3: safety related content for the Payload • Section 4: safety related content for the Platform • Section 5: Remote Safety related requirements and content • Section 6.4: Safety Interface data additions • Addition of Lite Adaptation Appendix (to provide high-level information for a CLIK Lite alternative to the CLIK standard configuration). • Updates to some mechanical and connector figures	

Contents

Figures

TABLES

1 SCOPE

1.1 PURPOSE

Rapid proliferation of Unmanned Aerial Systems (UAS) at all echelons of combat operations has introduced new capabilities for the Warfighter to deliver decisive overmatch and obtain freedom of maneuver across multiple domains. The pace of technology of UAS far exceeds any legacy military platform and associated acquisition lifecycle. UAS lifecycles will trend shorter with greater and more frequent technological platform upgrades. To minimize burden of integration at each platform upgrade, modular lethal payloads must have a common interface and architecture for attachment, power, communication, and safety critical commands. The desired end state is seamless interoperability of the UAS, the modular lethal payload, and the ground station controller.

Combat Capabilities Development Command (DEVCOM) Armaments Center (AC) is developing a platform agnostic, Picatinny Common Lethality Integration Kit (CLIK) to weaponize low-cost, attritable UAS. The purpose of this standard is to minimize the proliferation of physical and electronic interface standards on the UAS and the proliferation of physical and electronic interface standards on the payloads. The goal is to enable a soldier to move, mix and match different payloads on different platforms, without requiring a certification for each combination.

This document provides the base concepts, architectures, requirements, and overview for defining the required interoperability of Picatinny CLIK Ecosystem of packaged payloads. This document has been developed with existing reference architectures such as the Modular Payload Design Standard (MOD Payload) and small universal payload interface (sUPI) in mind. The intent is for this document to augment existing reference architectures with a focus on lethality integration and interoperability. Additionally, airworthiness and system safety certifications have influenced the standard with the intent to reduce burden associated with those certifications both to the user and to the developers.

1.2 SYSTEM OVERVIEW

This document specifies interoperability concepts, architecture, and requirements associated with integrating unmanned systems and lethal/nonlethal payloads. The capabilities addressed within this design standard relate primarily to lethal/nonlethal weapon payload interoperability with respect to payload integration, operation, control and status. To the degree possible, capabilities have been broken down into subsets to cover integration of specific interfaces/technologies associated with the integration of lethal payload technology.

A payload is a physical device that interfaces to the UAS. A payload can be similar in nature to other devices that are integrated on a UAS platform, but a payload is not required for native UAS capabilities. Frequently payloads are integrated into the platforms in a very specific way which isn’t always transferable to other platforms using physical, power, and data interfaces. CLIK will define where the boundary between platform and payload exists permitting rapid integration as well as interoperability. Modular and interchangeable payloads will enable warfighters to select the most efficient effect based on mission needs. Figure 1 depicts a notional diagram for the domain context that Picatinny CLIK supports.

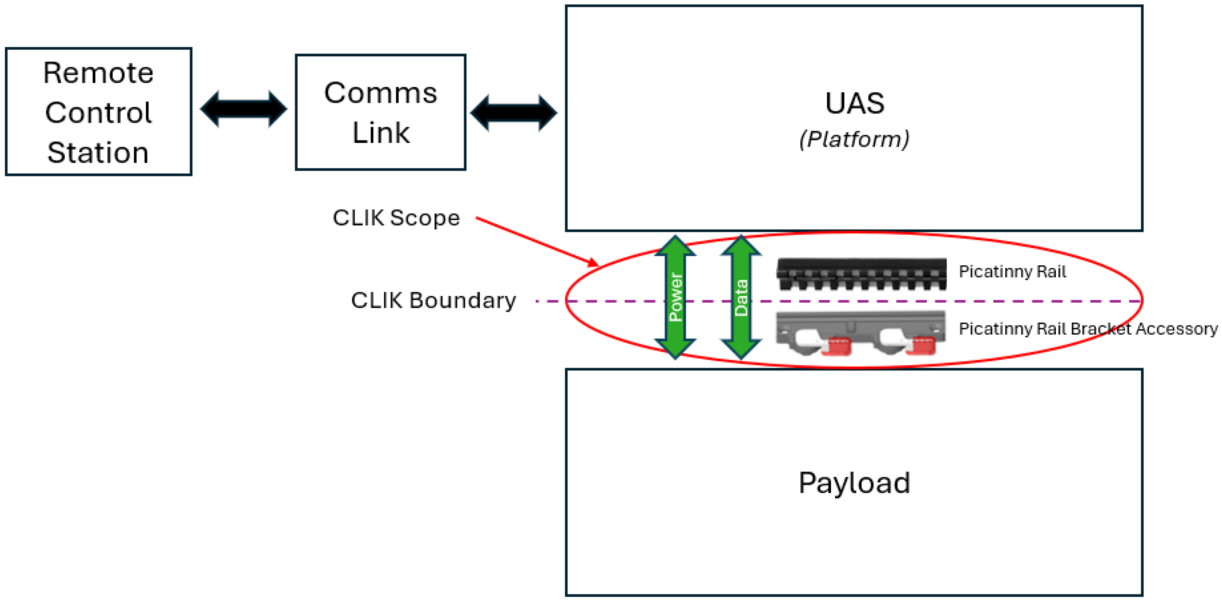


Figure 1 – Picatinny CLIK Domain Context

1.3 PICATINNY CLIK INTEGRATION LEVEL APPROACH

The Picatinny CLIK Ecosystem has a tiered (i.e. phased) approach to the capabilities it provides. There are four integration levels. However, at this time, the design standard is focused on integrating Picatinny CLIK to platforms in active prototype and experiment with DoD components and does not fully describe Integration Level 4, the highest level of integration. The intent of the integration level approach is to augment and assist Airworthiness configuration certifications in the future. Table 1 defines the requirements for attaining a level of CLIK integration.

Integration Level 1: In this initial phase, Picatinny CLIK provides a physical interface only for Payload System connecting to the rail(s). The Payload System has all the subsystems (which may include communications, power, sensors, command and control) required for full operation with no dependence on the Platform including its own Ground Control Station.

Integration Level 2: In this second level, the Picatinny CLIK provides a power interface between the Platform and Payload systems that can be used. The Payload system would still maintain its own Ground Station.

Integration Level 3: In this third level, the Picatinny CLIK adds interfaces for communications (i.e. Radio, Command and Control (C2)) between the Platform System to the Payload System that can be used. The payload C2 software could be integrated on the Platform Ground Control Station. A remote safety device and its communication with the platform is also defined to permit the safe transmission of safety signal states to the platform.

Integration Level 4: In this fourth phase (which is not fully described in this design standard at this time), the Picatinny CLIK adds interfaces for sensors between the platform to the payload system that can be used. This would be the full integration and interoperability of the Payload and Platform Systems. The Payload system may contain its own sensors as desired (to increase accuracy or supplement Platform sensors).

Integration Level	Platform System	Payload System	Remote Safety
1	Section 4.1	Section 3.1	N/A
2	Sections 4.1, 4.2.1.1, 4.2.3.1, 4.2.3.2, 4.2.4, 4.3	Sections 3.1, 3.2.1, 3.2.2, 3.2.3, 3.2,4	N/A
3	Sections 4.1, 4.2, 4.2.1.1, 4.2.2.1, 4.2.2.2, 4.2.3, 4.3, 6.1, 6.3, 6.4, 6.5	Sections 3.1, 3.2, 3.2.1, 3.2.2, 3.2.3, 3.2,4, 6.1, 6.2	Sections 5, 6.4
4	To Be Defined in Future Revision		

Table 1: Integration Level Attainment

2 APPLICABLE DOCUMENTS

2.1 GOVERNMENT DOCUMENTS

MIL-STD-1913		Dimensioning of Accessory Mounting Rail for Small Arms Weapons
MIL-HDBK-338B		Electronic Reliability Design Handbook
MIL-STD-464D		Electromagnetic Environmental Effects Requirements for Systems
MIL-STD-1472H		Human Engineering
MIL-STD-882E		Systems Safety
MIL-STD-130N		Identification Marking of U.S. Military Property
MOD Payload		A Modular Payload Design Standard Revision 6.1
sUPI Platform Brief		Distro A sUPI Platform Interface ICD
sUPI Payload Brief		Distro A sUPI Payload Interface ICD
sUPI TDP		Distro C Detailed Platform and Payload Interface Technical Data Package

Table 2: Government Documents

2.2 NON-GOVERNMENT DOCUMENTS

IEEE 802.3u		IEEE Standard for 100BASE-T (Fast Ethernet) and 100BASE-FX (Fast Ethernet over Fiber)
IETF RFC 768		User Datagram Protocol (UDP)
IETF RFC 4632		Classless Inter-Domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan
EIA/TIA-232-F		Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange
ANSI X3.4-1986		Coded Character Sets – 7-Bit American National Standard Code for Information Interchange (7-bit ASCII)
IEC 60747-5-5		Semiconductor devices – Part 5-5: Optoelectronic devices – Photocouplers
IEC 60747-17		Semiconductor devices – Part 17: Magnetic and Capacitive Coupler for Basic and Reinforced Insulation
IEC 62541-15		OPC Unified Architecture – Part 15: Safety
IEC 61508		Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems
IEC 61784-3		Industrial communication networks – Profiles – Part 3: Functional safety fieldbuses – General rules and profile definitions
ITU-T V.41		International Telecommunication Union, ITU-T Recommendation V.41: Code-independent error-control system, 1988.

Table 3: Non-Government Documents

3 PAYLOAD REQUIREMENTS

The following sections specify the mechanical, electrical, and protocol requirements for a Picatinny CLIK compliant payload. The majority of platforms will be able to support the structural, power and signal interface outlined for CLIK Payloads. Some platforms may require alternative mounting and electrical interfaces due to their size and mission profile. The CLIK Lite adaptation (modeled based on sUPI) is the alternative for platforms constrained by low capacity for payload size, weight, and power needs (see Appendix A). This interface is suitable for Short Range Reconnaissance (SRR) and Purpose Built Attritable System (PBAS) type aircraft with Picatinny CLIK compliant payloads. The initial CLIK Lite and ecosystem components have been transitioned to DEVCOM AC for incorporation into Picatinny CLIK and contained in Appendix A.

3.1 MOUNTING/STRUCTURAL INTERFACE

The physical interface requirements listed in this document describe which standards should be used. This does not ensure that a particular mounting(s) will support a payload, provide stable mounting, or provide the proper vehicle dynamic characteristics. It is the responsibility of the lead system integrator and payload developers to ensure that these characteristics are met, and pass testing, in a particular implementation.

3.1.1 PICATINNY RAIL

CLIK payloads shall be capable of mounting to a single rail or multiple rails in accordance with MIL-STD-1913. Connection coupler(s) of varying length not less than 5” in length and then 5” on center (OC) from side to side up to the maximum width of the UAS mounting structure. The number of rails required to mount a payload is determined by the payload developers for proper payload mounting and spacing considerations. The payload must go through proper analysis of anticipated flight profiles and maneuvers. This data will need to be provided so that integrators can make informed decisions when integrating a payload onto a platform.

3.1.2 PICATINNY RAIL BRACKET ACCESSORY

The Picatinny CLIK interface brackets shall be compatible with MIL-STD-1913 type accessories and ideally toolless in design. If tools are required, a standard issue military multitool, such as a Leatherman pocket survivor tool, is maximally allowed. It must have a redundancy feature for secondary means of attachment to ensure adequate clamp strength for static and dynamic loads, positive control and prevent dislodging or detaching during standard flight conditions. A notional example of a toolless bracket accessory is shown in Figure 2.

The Picatinny Rail Bracket Accessory shall have a maximum length of five inches.

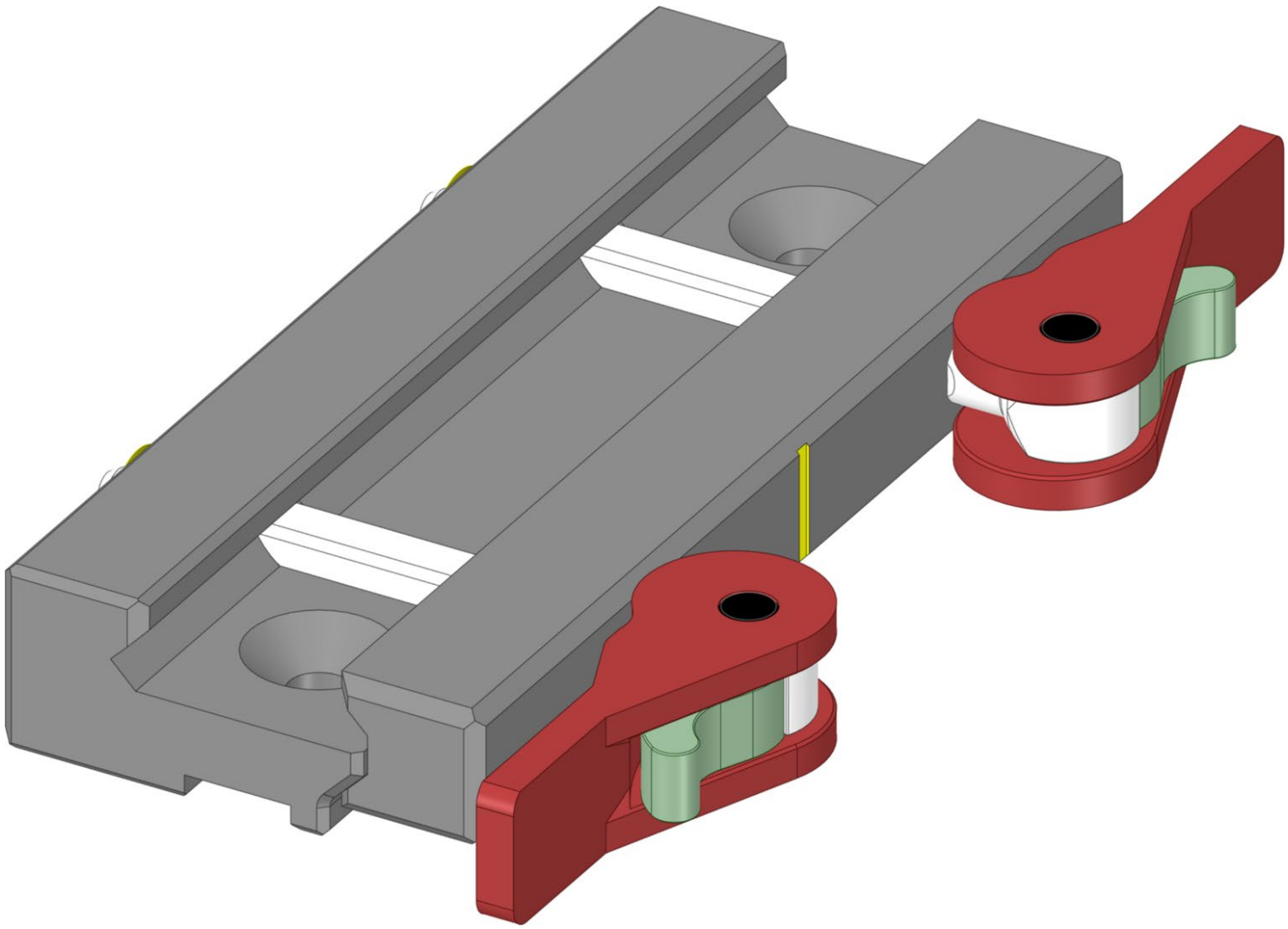


Figure 2 – Notional Bracket Accessory

3.1.3 PICATINNY RAIL BRACKET MARKING IDENTIFIER

The minimum height of a Picatinny Rail is governed by MIL-STD-1913. The maximum height of the rail space claim on the payload is a 0.4" slot opening along the entire length to allow the payload bracket to be placed anywhere along the rail (see Figure 3).

3.1.3.1 MARKING

The mounting bracket shall incorporate a marking indicating the payload's center of gravity (CG). This will permit integrators to align payload CG with the platform CG. For Platform marking requirements, see 4.1.1.

3.1.3.2 WEIGHT

The physical properties of the payload and mounting system must be described.

The payload connection system must withstand a maximum 5G acceleration in all directions during platform maneuvers.

3.1.4 CLIK UNIT

A CLIK Unit defines the base logical requirements for interoperability between a Payload System and a UAS Platform. Platforms possess finite resources related to weight, size, power, and data throughput. A CLIK Unit represents the expected boundary conditions for a single Payload System. While the physical dimensions of a CLIK Unit are expected to evolve as this design standard expands, the core principle remains consistent: to ensure interoperability.

1 CLIK Unit (1CU) nominally defined as 4.25" Wide, 12.75" Long and 1" High. For a payload less than 4.25" Wide and 12.75" Long use the following notation: 1 CU, H". For payloads that exceed the dimensions of 1 CLIK Unit must be labeled as an array of CLIK Units: W x L CU, H". L defines the space claim along the length of the rail. W defines the space claim perpendicular to the rail and H defines the space away from the surface of the rail.

The coordinate system that all payload mass properties should use is the controlled surface at the base of the 1913 rail (ref. 0.367") for the XY plane (see Figure 3). For most platforms, the minimum 0.367" will be used and the coordinate system will be on the bottom plane of the aircraft to aid in airworthiness analysis. +Z points towards the mounting surface, and -Z points away. See 4.1 for more information on reference axes.

The payload shall have a clearance of 1.0"x0.40" minimum center aligned at the top of the payload container as seen in Figure 6 **t**o allow appropriate clearance for rail.

The required space claim for soldier access to notional bracket attachment mechanism is depicted in an example configuration in Figure 3 (will vary depending upon specific rail attachment method, see Section 3.1.6).

The payload shall come up with a launch and recovery platform or integrate a mechanism to permit launch and recovery into the payload if the payload volume is larger than the platform can support (i.e. provide legs or provisions for payload to sit on ground if platform legs are too short or nonexistent).

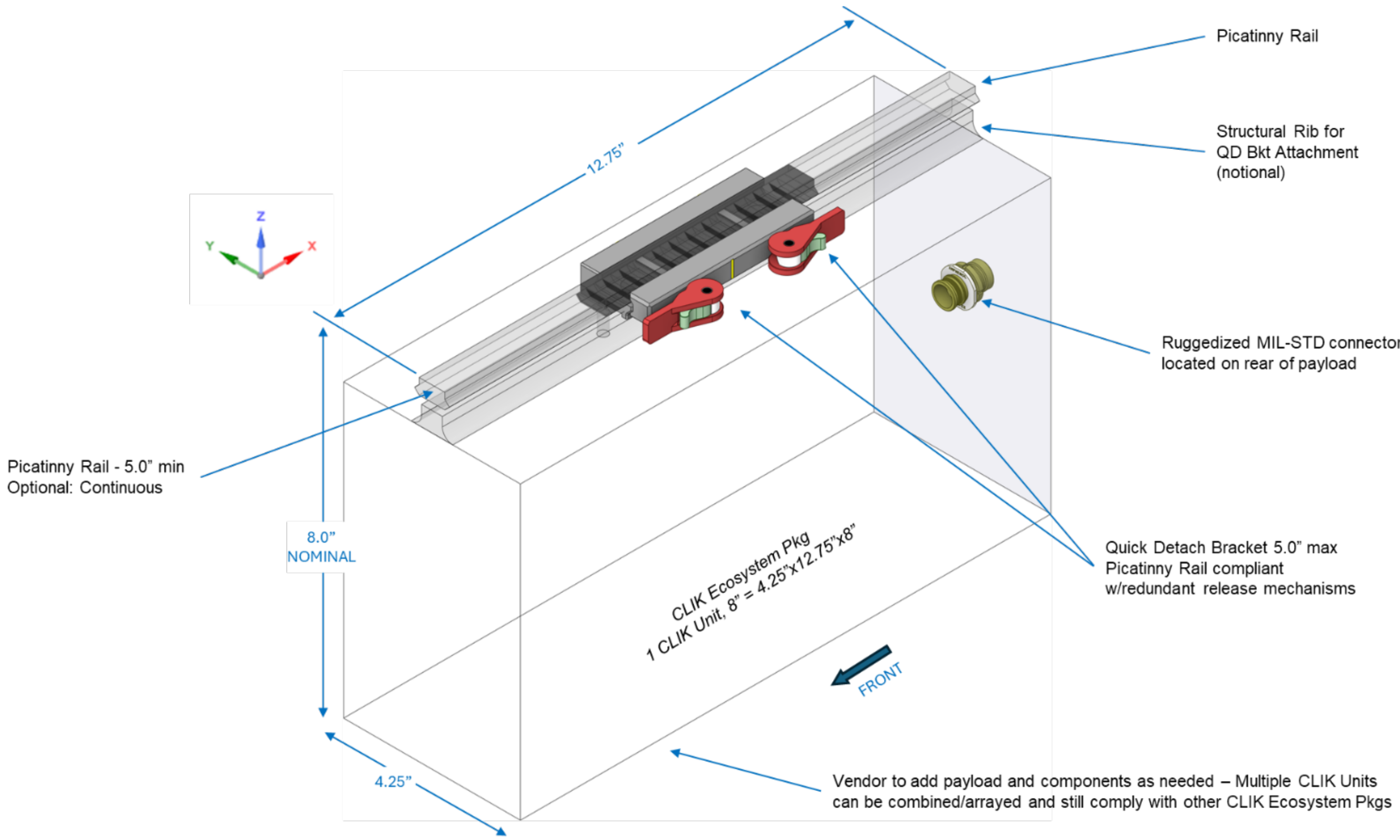


Figure 3 - Picatinny CLIK 1 CU,8" Nominal Space Claim Example

Figure 4 shows a notional example of a CLIK Unit structure.

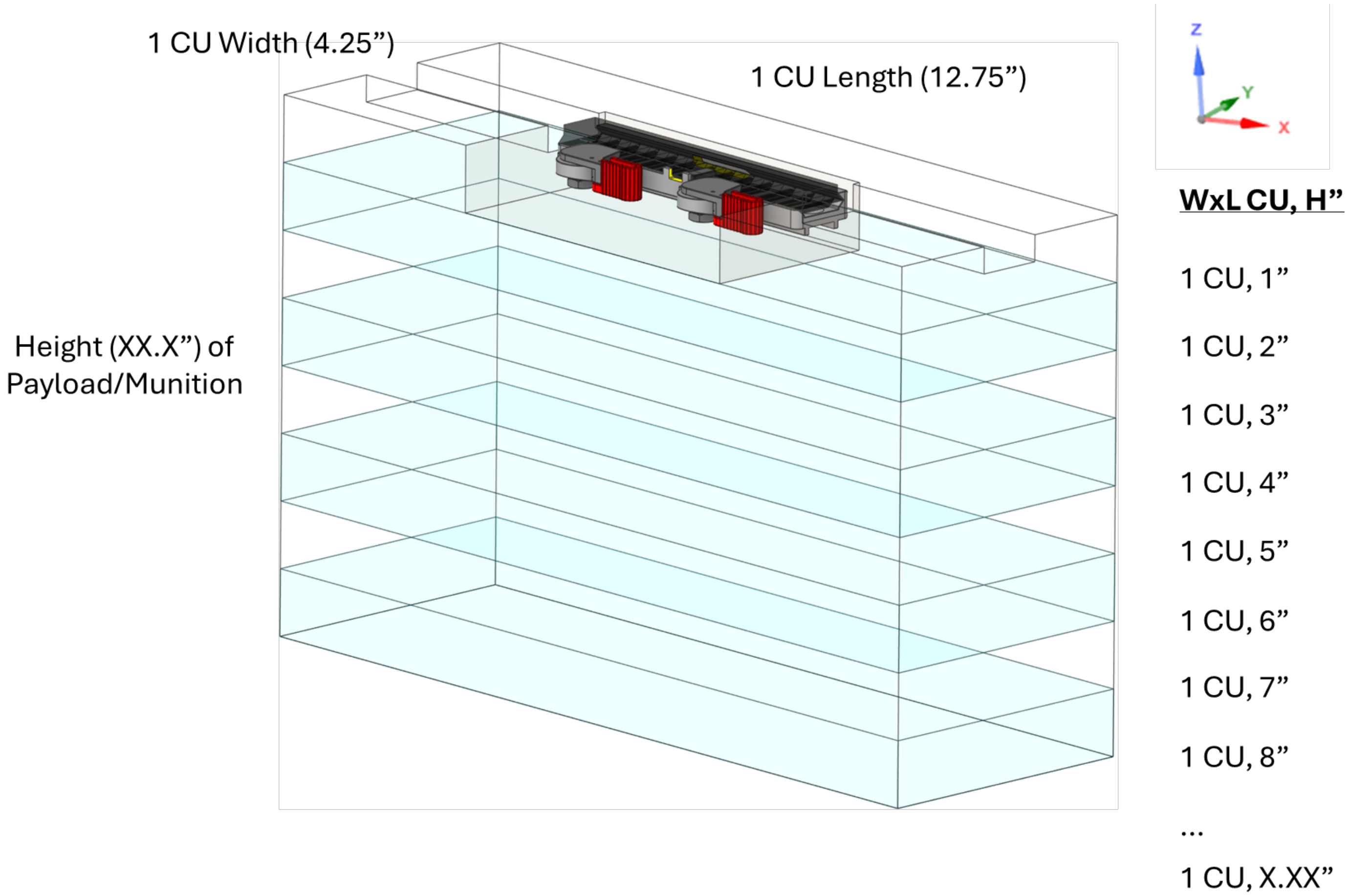


Figure 4 – CLIK Unit Example for payload height in inches

3.1.5 PICATINNY CLIK ECOSYSTEM

3.1.5.1 SINGLE PICATINNY CLIK PAYLOAD

A CLIK Payload operating in standalone has greater flexibility of deployment than one that must operate with other CLIK Payloads (notional examples are shown in Figure 5 and Figure 6).

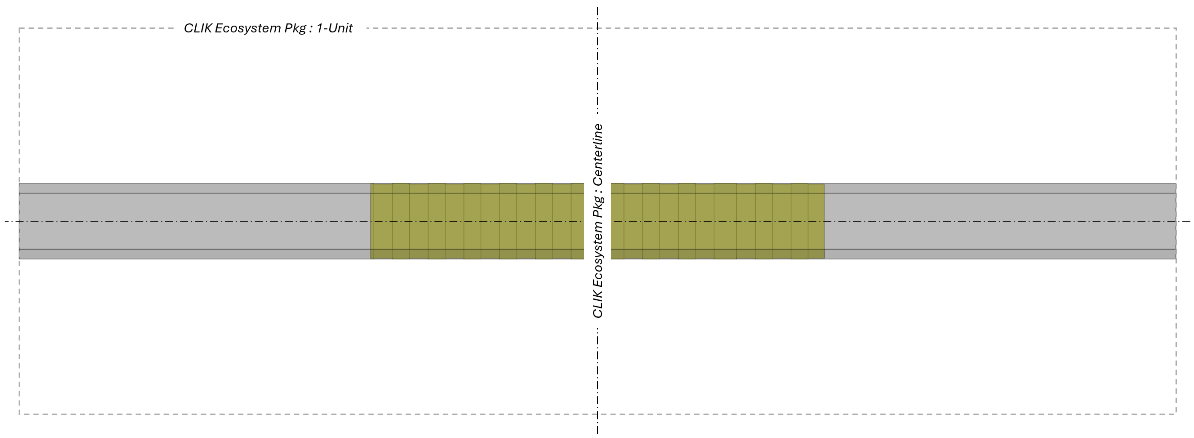


Figure 5 – Single Picatinny CLIK Ecosystem Package Unit

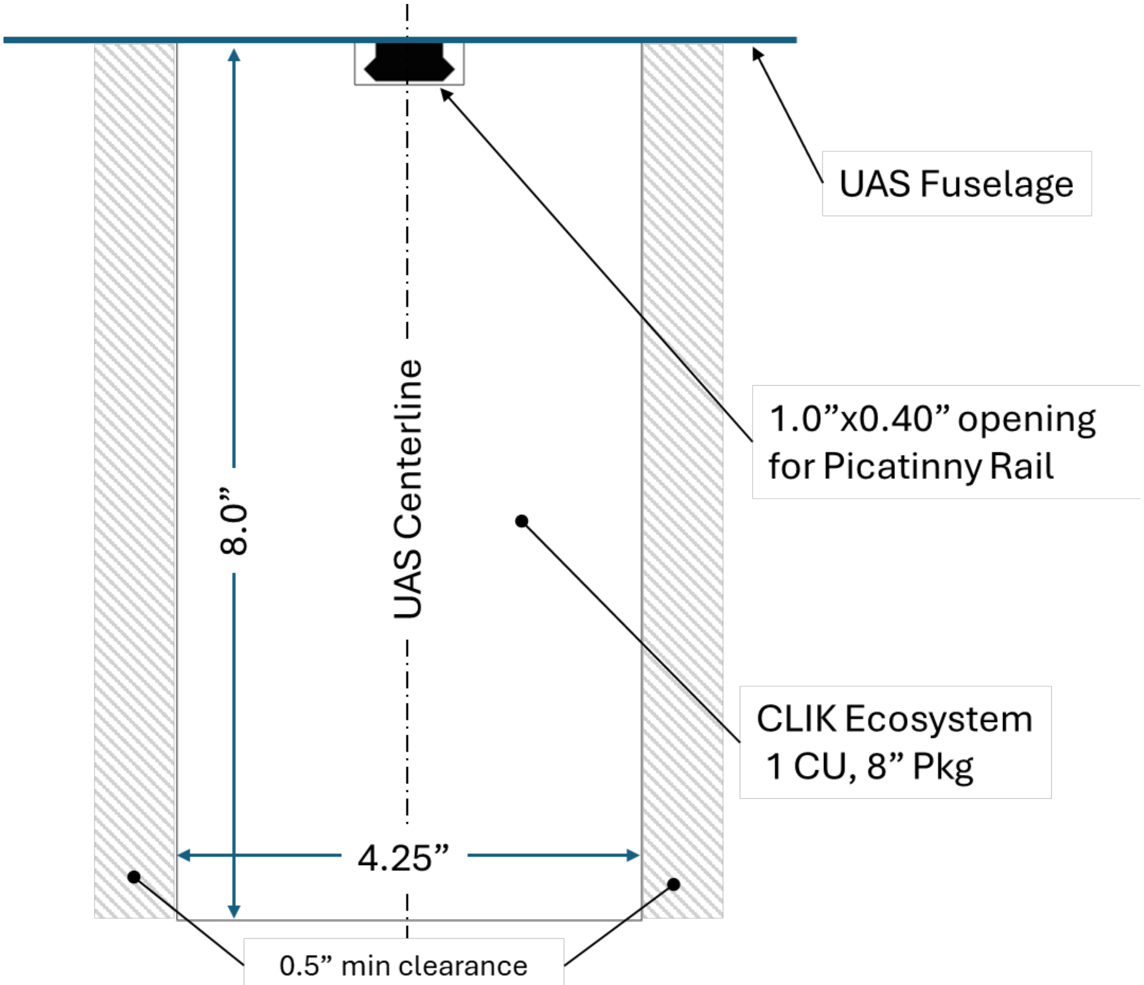


Figure 6 – Payload Clearances

3.1.5.2 MULTI RAIL ADAPTER

When a Picatinny CLIK payload exceeds the platform’s single rail attachment capability, a multiple rail approach is allowed to distribute loads across a greater number of rails.

3.1.5.3 STANDARD GRID LAYOUT FOR MULTIPLE PICATINNY CLIK ARRAY LAYOUT

A CLIK Array is defined as a configuration consisting of two or more CLIK Payloads mounted in a defined grid pattern (suggested grid pattern; up to UAS to analyze whether they can conform to suggested or need an alternate configuration). This arrangement enables the integration of larger payloads or a compilation of smaller payloads. The figures that follow showcase notional examples of CLIK array configurations (Figure 7, Figure 8, Figure 9, Figure 10, and Figure 11).

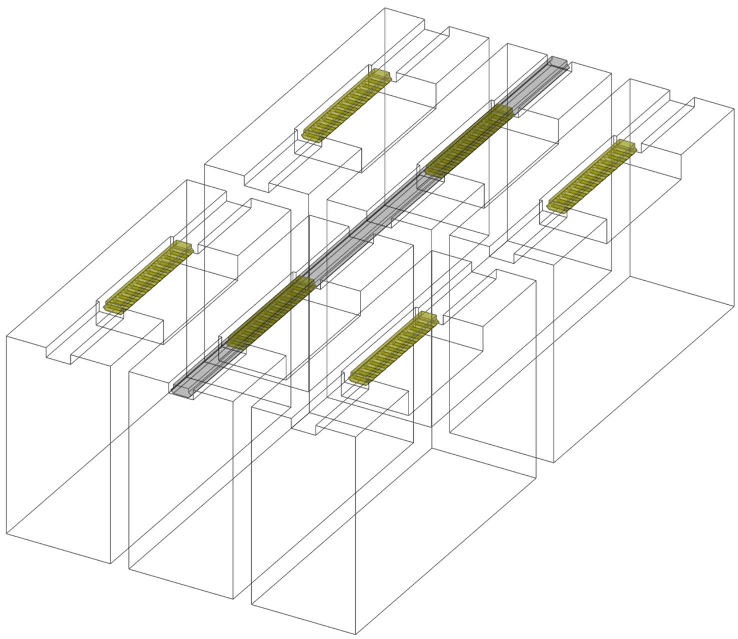


Figure 7 – Mult-Picatinny CLIK Array (2x3CU)

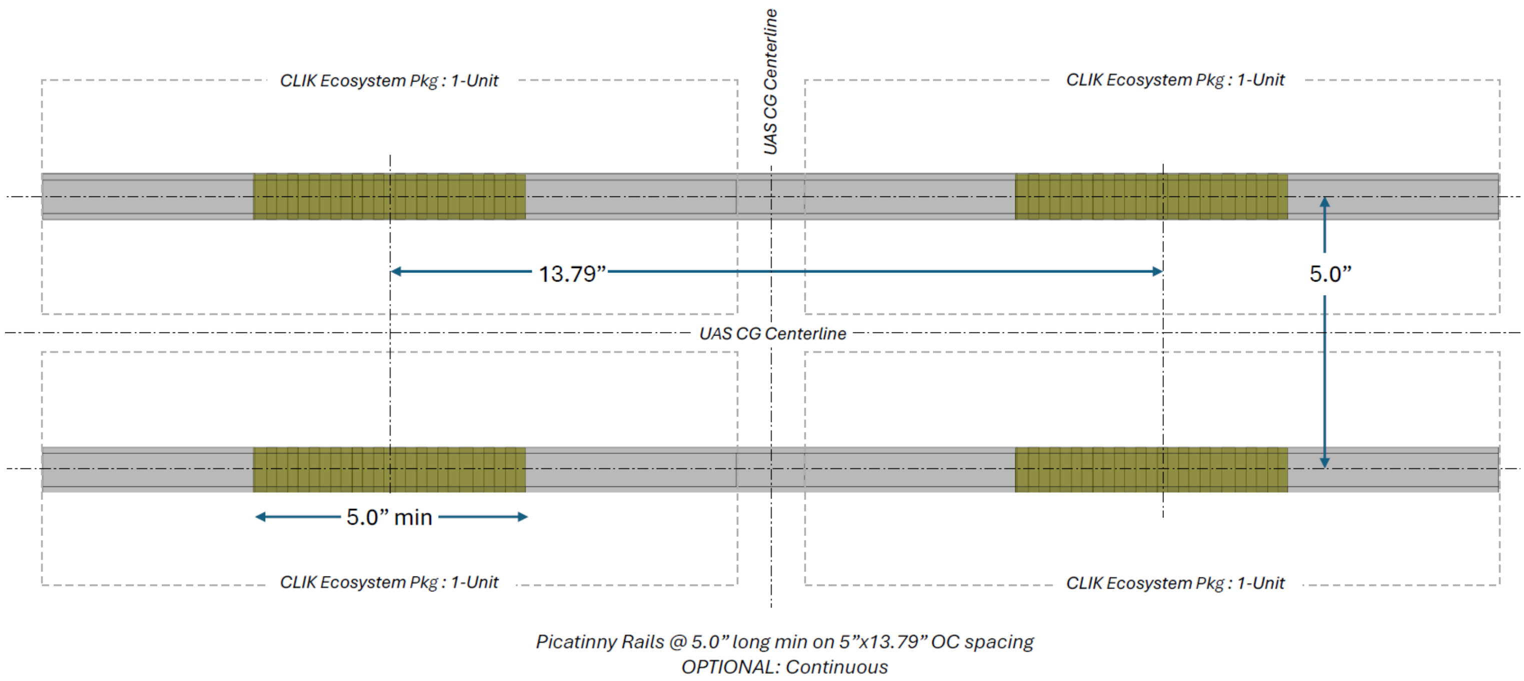


Figure 8 - Picatinny CLIK Ecosystem Package Even Array (2x2CU)

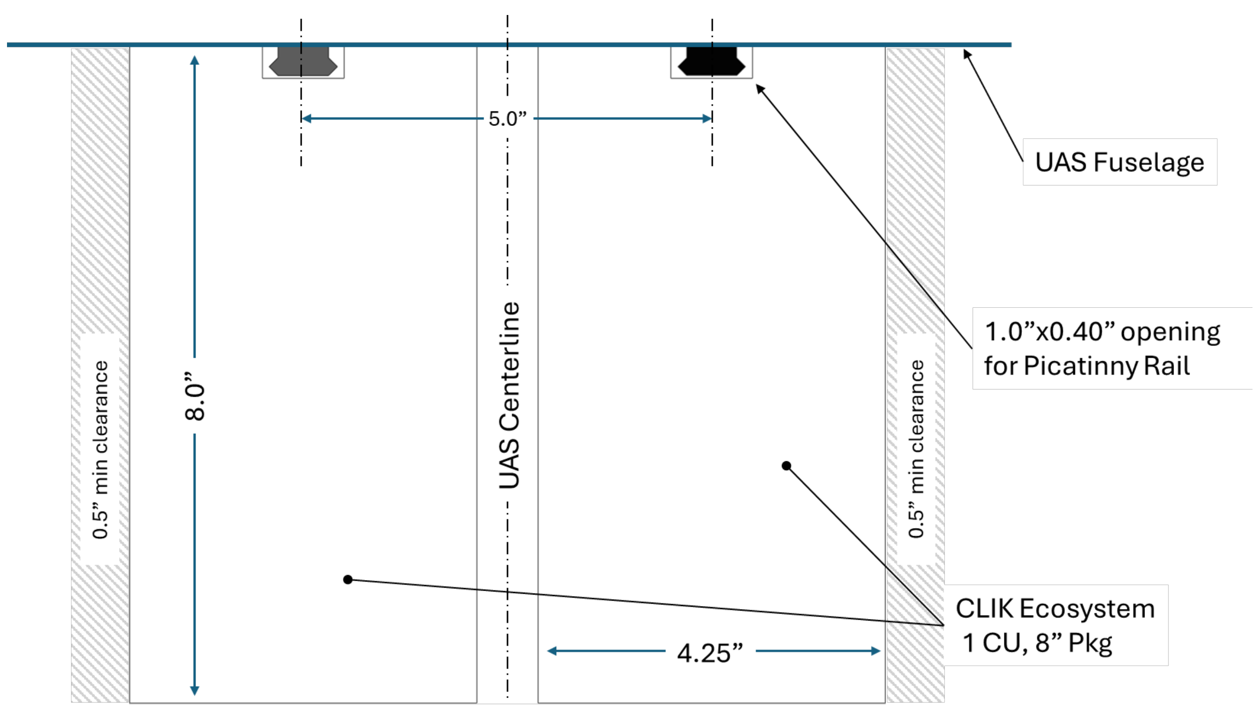


Figure 9 – Dual Rail Required Clearances

UAS unladen center of gravity will always be marked (see Figure 10) on the platform to address mounting of a single Picatinny CLIK Pkg on a UAS which can mount multiple numbers of Picatinny CLIK payloads so that a balanced load reference location can be identified.

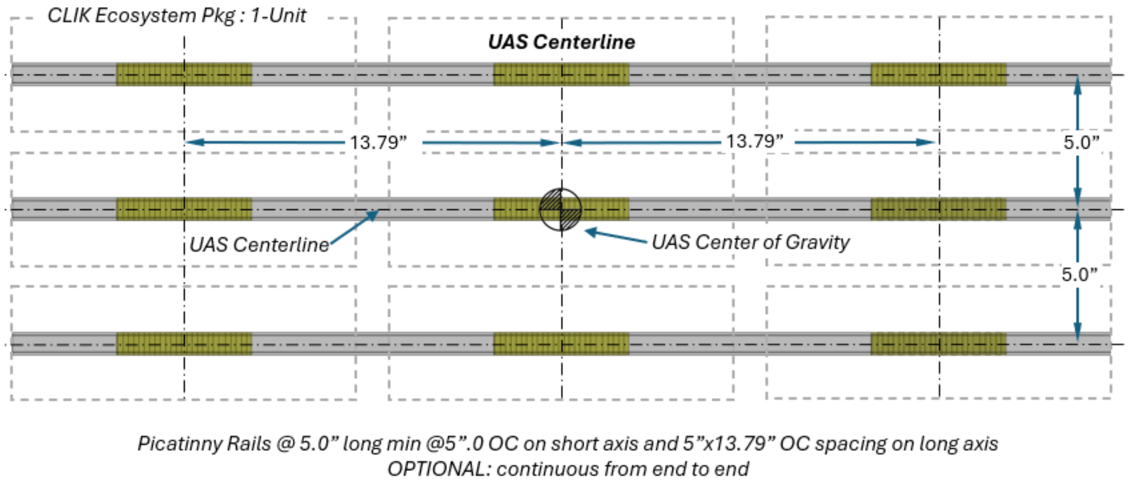


Figure 10 - Picatinny CLIK Ecosystem Pkg Odd Array (3x3CU)

Array configuration when using a barrel type weapon in conjunction with additional Picatinny CLIK pkgs (see Figure 11).

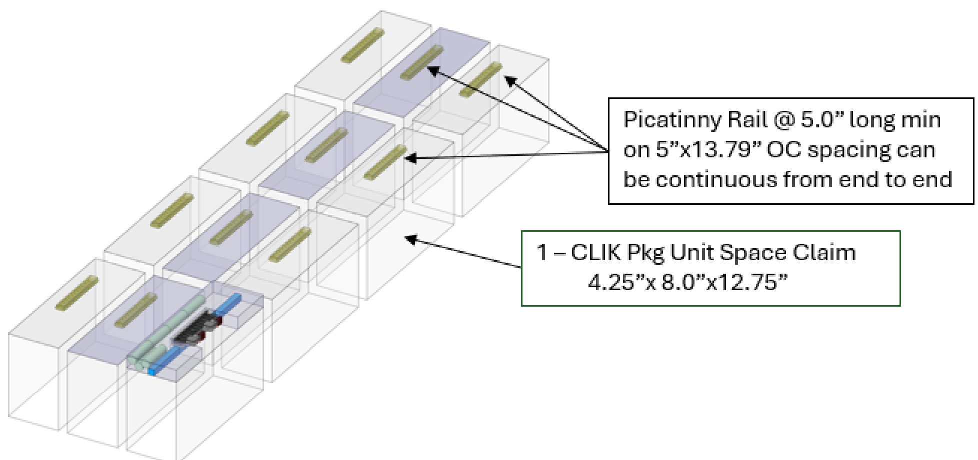


Figure 11 - Picatinny Rail layout when carrying a single long item "as shown in blue"

3.1.6 PAYLOAD HUMAN FACTORS

The Payload shall be installable onto the Picatinny CLIK System by the Soldier without the need for special tools.

The Payload shall be removable from the Picatinny CLIK System by the Soldier without the need for special tools.

All System operations regarding interfacing the Payload System to the Picatinny CLIK System performed by the Soldier shall be accomplishable when the Soldier is wearing tactical gloves.

If tools are required, a standard issue military multitool, such as a Leatherman pocket survivor tool, is maximally allowed.

3.1.7 IDENTIFICATION MARKINGS

The Payload shall display a permanent identification plate with human readable text and Machine-Readable Information (MRI) in accordance with MIL-STD-130N. The identification plate shall include the following minimum information: manufacturer, part number, unique serial number, maximum payload weight in pounds to one decimal as MPW: XX.X, and maximum payload current in Amperes to one decimal as MPC: XX.X (see 3.2.4).

3.2 POWER AND SIGNAL INTERFACE

The majority of CLIK Platforms and Payloads will be able to support or require the power and signals outlined in the following interface. Smaller UAS such as PBAS and SRR may implement the CLIK Lite addendum. It is expected that a common ecosystem component will be developed to bridge the use of CLIK Lite payloads with the outlined interface.

3.2.1 PAYLOAD CONNECTOR

A Picatinny CLIK payload shall integrate a single connector from the currently available options shown in Table 4 or a connector with identical form, fit, and function as detailed in sub sections below.

<table>
<tr><th>Series</th><th>Class</th><th>Shell Style</th><th>Material/Finish</th><th>Arrangement</th><th>Contacts</th><th>Keying</th></tr>
<tr><td>801</td><td rowspan="2">-009</td><td>-01</td><td rowspan="2">NF</td><td rowspan="4">9-19</td><td rowspan="4">P</td><td rowspan="4">A</td></tr>
<tr><td rowspan="2">2M801</td><td rowspan="2">-02</td></tr>
<tr><td rowspan="2">-010</td><td rowspan="2">ZNU</td></tr>
<tr><td>MD801</td><td>-07</td></tr>
</table>

Table 4: Approved Payload Connectors

3.2.1.1 SERIES

Picatinny CLIK Payload Connector shall be of the following Series options shown in Table 5.

Manufacturer	CAGE Code	Product Line	Description
Glenair, Inc.	06324	801	Mighty Mouse Double Start Acme Thread
Amphenol Corp.	77820	2M801	Micro-Miniature Dual-Start Acme Thread
Cinch Connectivity Solution Inc.	34078	MD801	MD801 Dual-Start Acme Thread

Table 5: Approved Manufacturer's Connectors Series

3.2.1.2 CLASS

Picatinny CLIK Payload Connector shall be of Receptacle Class Series options shown in Table 6.

Series	Description
-009	Receptacle with Banding Platform
-010	Receptacle with Accessory Threads

Table 6: Approved Connector Class Series

3.2.1.3 SHELL STYLE (MOUNTING)

The approved manufacturers provide various options for how the connectors mount. Figure 12 shows mounting options typically available from the listed manufacturers.

Figure 12: Receptacle Connector Typical Mounting Options

Picatinny CLIK Payload developers can select the specific option in Table 7 that best suits their application.

Shell Style	Description
-01	In-Line
-02	Square Flange
-07	Jam Nut

Table 7: Shell Style Options

3.2.1.4 MATERIAL/FINISH

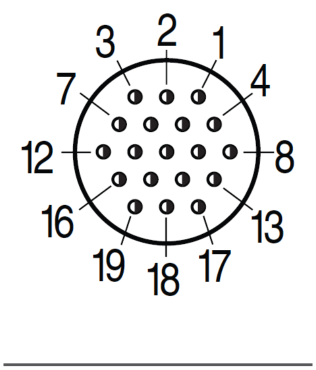
The finish type shall be selected from the options shown in Table 8.

Material/Finish	Description
NF	Aluminum/Cadmium with Olive Drab Chromate
ZNU	Aluminum/Black Zinc Nickel over Electroless Nickel

Table 8: Approved Connector Material/Finish

3.2.1.5 ARRANGEMENT

Picatinny CLIK Payload Connector shall be of Shell Size/Insert Arrangement of 9-19. Refer to Figure 13 for Contact Arrangement. Note that this image represents the front face of male (pin) contacts. Female (socket) contacts will position pins in a mirror position to the pin format shown.



9-19

Figure 13: Payload Connector Insert Arrangement

3.2.1.6 CONTACTS

Picatinny CLIK Payload Connector shall use Male (Pin) Contacts of Size #23.

3.2.1.7 KEYING

Picatinny CLIK Payload Connector shall be keyed in the A Configuration as shown in Figure 14.

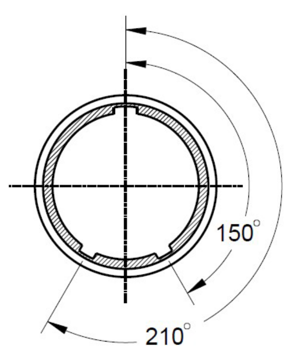


Figure 14: Connector Keying

3.2.2 CONNECTOR PLACEMENT

The connector will typically be located on the back plane of the payload. The specific location will be determined by the platform developer. A notional example is provided in Figure 15.

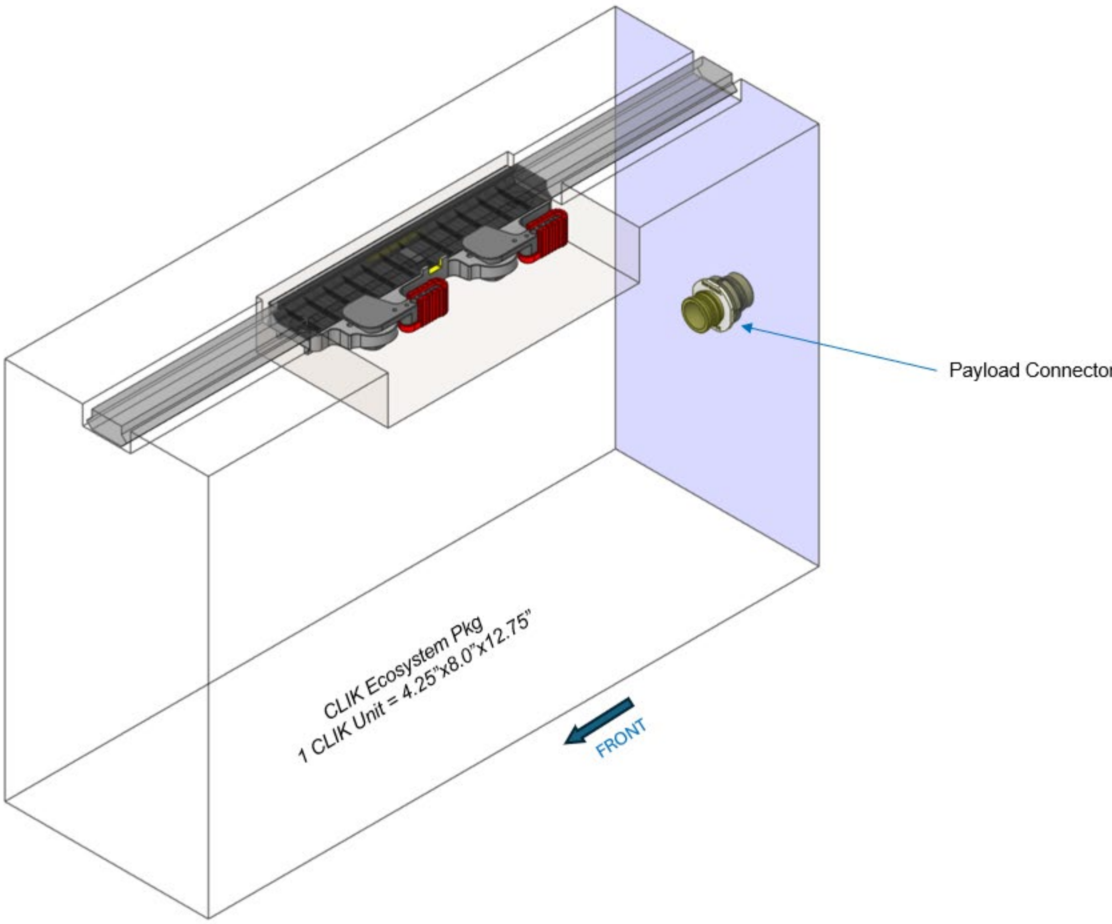


Figure 15: Payload Connector location (notionally on payload backplane)

3.2.3 CONNECTOR PINOUT

A Picatinny CLIK payload shall implement its connector pinout as described in Table 9. All signals are named from the payload perspective such that an Rx signal is transmitted from the platform and received by the payload. Pins marked as Reserved are retained for future expansion of the Picatinny CLIK design standard and shall not be used by the payload for other purposes. The connector's 9-19 insert arrangement as seen from the front face of pin inserts is shown in Figure 13.

Pin	Name	Description	Reference
1	RS232 Tx	Serial Port.	3.2.6
2	RS232 Rx		
3	RS232 Return	Return path for RS232.	
4	Ethernet Tx+	Ethernet Port.	3.2.5
5	Ethernet Tx-		
6	Ethernet Rx+		
7	Ethernet Rx-		
8	Power In	Power in from the platform	3.2.4
9	Power In		
10	Chassis	Chassis bonding connection	3.2.10
11	Power Return	Power return to the platform	3.2.4
12	Power Return		
13	Safety Enable	Safety signals and dedicated return path.	3.2.8
14	Safety Execute		
15	Safety Return		
16	Loopback	Loopback signal for payload detection.	3.2.9
17	Reserved	Pins reserved for future Picatinny CLIK expansion. Not to be used.	3.2.3
18	Reserved		
19	Reserved		

Table 9: Payload Connector Pinout

3.2.4 POWER

A Picatinny CLIK payload shall operate from a supplied 28 Volt (V) Direct Current (DC) power source. The payload shall remain operational under voltage fluctuation of ±5% from nominal 28VDC. The maximum current draw permitted through the Picatinny CLIK connector shall be limited to 10 Amperes.

Picatinny CLIK payloads shall determine their Maximum Possible Current (MPC) draw. Picatinny CLIK payloads shall implement overcurrent protection to ensure the payload does not draw more than its rated MPC from the platform under normal operations and possible failure modes. A Picatinny CLIK payload shall not incur damage due to being connected or disconnected from a platform with power present.

A Picatinny CLIK payload shall not permit current to flow back to the platform unless the payload is specifically purposed for energy storage. A Picatinny CLIK payload designed for energy storage shall not permit more than 10A to flow through its Picatinny CLIK payload connector and shall maintain operation within the permitted 28V tolerance.

3.2.5 ETHERNET

A Picatinny CLIK payload shall optionally implement an Ethernet interface in accordance with Institute of Electrical and Electronics Engineers (IEEE) 802.3u 100BASE-TX. The Internet Protocol (IP) addressing scheme is left to the system integrator. A Picatinny CLIK payload implementing Ethernet shall permit the interface to be enabled and disabled with the default state at power on being disabled.

3.2.6 SERIAL

A Picatinny CLIK payload shall implement a serial interface IAW TIA/EIA-232-F. The recommended configuration for this port is 115,200 bps, 8 data bits, 1 stop bit, no parity, and no flow control. Deviation from the recommended configuration would require coordination with the platform providing this channel. Serial Tx and Rx signals shall be referenced to RS232 Return. The RS232 Return is permitted to be electrically connected to Power Return in the payload only.

A Picatinny CLIK payload shall implement the Payload Serial Protocol as defined in 6.2 over the serial interface.

3.2.7 FUNCTION CHANNELS

A Picatinny CLIK payload may implement one or more function channels. Function channels are used to perform payload operations, often in conjunction with the CLIK safety signals. Payloads shall assign each of its function channels a unique numeric value starting with 1 and incrementing sequentially.

Function channels shall exist in a defined state as defined in Table 10. A Function Channel's current state shall be acquired via the Get Payload Status message in 6.2.1. A function shall accept commands to switch between the Not Activated and Activated states via the Set Function State command specified in 6.2.5. If a Set Function State command is not valid in the current function state a "Not valid in current state" error code shall be returned. Unless explicitly stated, the acceptance and behavior associated with receipt of a Set Function State, as well as the logic criteria for transitioning between states, shall be payload specific and is not dictated by this document. Note that qualifying a payload for a particular application may necessitate specific logical interactions between function channels, safety signals, and other safety mechanisms.

State Name	ID	Description
Unavailable	0	Function not available (i.e. out of ammunition)
Faulted	1	Function in a faulted state and cannot be used
Not Activated	2	Function is available but has not received activate command
Activated	3	Function has been activated and awaiting further conditions
Moving	4	Payload function is causing something to actively move
Arming	5	Payload function is actively moving towards an Armed state
Armed	6	Payload function is currently Armed
Firing	7	Payload function is actively moving towards a Fired state
Fired	8	Firing associated with function has completed
Completed	9	Operation associated with function has completed
Failed	10	Previous operation associated with function has failed
Reserved	11-127	Reserved for future use by CLIK
Available	128-255	Available for payload specific use.

Table 10: Function Channel States

Function channels can be used for controlling lethal functions as well as for other operations. For example, an energy storage payload could be commanded to the Activated state to enable its output and could then transition to the Unavailable state once its energy was depleted with the Safety Enable and Safety Execute signals never being utilized and the remaining function states never being set. For another example, imagine a munition dropper payload that requires an actuated motion to permit reloading. Its "Drop" function reports being in the Fired state, indicating that the munition has been dropped successfully. A "Reload" function, when activated, transitions the function from the Not Activated state to the Moving state and then back to the Not Activated state when the operation is completed and the new munition is loaded. This operation, when completed, could cause the "Drop" function to transition from the Fired to the Not Activated state.

3.2.8 SAFETY SIGNALS

A Picatinny CLIK payload shall monitor the Safety Enable and Safety Execute signals provided by the platform as defined in 4.2.2.3. A false value shall be interpreted as a signal with a voltage of less than or equal to 1.5V. A true value shall be interpreted as a voltage of greater than or equal to 3.5V. A Picatinny CLIK payload shall maintain the isolation requirements specified in 4.2.2.3. Safety signal current draw shall not exceed 5mA for each signal.

The safety signals, combined with the ability to enable one or more function channels, provides the ability for a remote operator to select independent payloads (and their functions) via software, and then selectively cause simultaneous actions to occur via activation of the global safety signals. Note that the safety functionalities provided by CLIK are a component of a payload's overall safety but are not exhaustive. Some payloads may require additional circuits, mechanisms, and/or commands to operate safely. The specific methods used to implement the full payload functionality are left to the developer, but a notional example is depicted in Figure 16 where the 28V power needed to execute a function is gated by several switching devices in series, thus requiring that each be enabled for the function to execute.

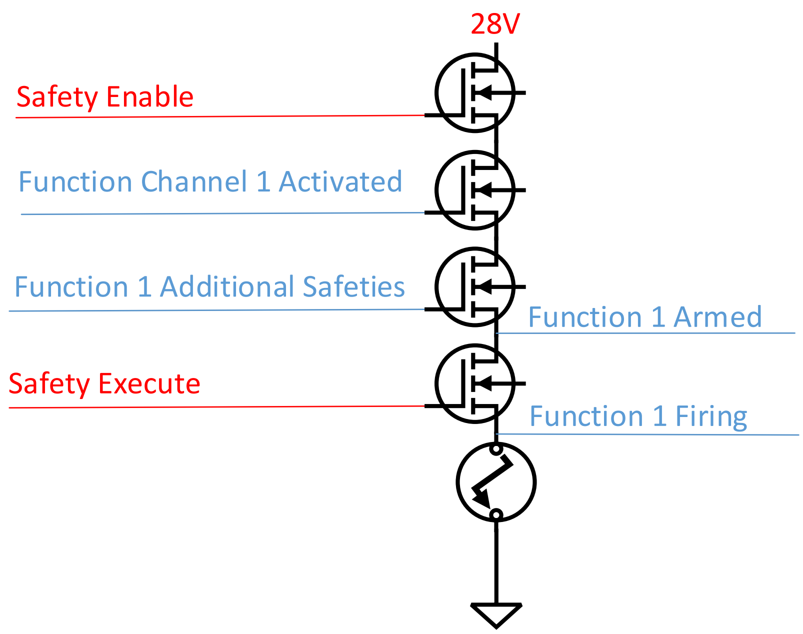


Figure 16 – Notional Payload Safety Circuit

3.2.9 LOOPBACK

A Picatinny CLIK payload shall connect the Loopback signal from the platform to Power Return with less than 1 Ohm resistance.

3.2.10 CHASSIS

A Picatinny CLIK payload shall electrically bond any conductive components of the payloads structure, shielding, connector body, and Chassis Signal pin. A Picatinny CLIK payload shall maintain electrical isolation between all CLIK defined electrical signals and the payload Chassis Signal.

4 PLATFORM REQUIREMENTS

4.1 STRUCTURAL INTERFACE

The Picatinny CLIK platform shall identify whether it is using node (single attachment point), array (multiple discrete attachment points), or slide (infinite attachment points) structural interface.

Platforms shall provide documentation on the allowable payload weights, and center of mass locations based on their structural analysis.

Platforms should consider their sensitivity to payload weight and center of mass when choosing a structural interface type. Node interfaces are the least adjustable and should be used for insensitive platforms while slide interfaces give the most amount of adjustment. Platforms should also consider their own sensitivity to vibrations, shock loads, and sudden changes in payload mass when designing their payload structural interfaces.

Picatinny CLIK structural interfaces shall align their mounting surface normal to one of the platforms primary yaw, pitch or roll axes where roll is rotation about axis X, the forward direction of flight, yaw is rotation about axis Z, antiparallel to the gravity vector, and pitch is rotation about axis Y, outboard from the platform CG, perpendicular to X and Z using right handed convention.

Platforms shall provide adequate ground clearance for mounted payloads, as defined in 3.1.4. Use of a takeoff platform is permitted to achieve adequate clearance.

The Picatinny CLIK platform array interface shall utilize a Picatinny rail identified in 3.1.1. They shall meet the dimensional requirements of MIL-STD-1913.

The Picatinny CLIK platform slide interface is yet to be determined.

A Picatinny CLIK platform of integration level 2 or higher shall provide access to one or more Picatinny CLIK connectors to permit mating to an associated payload. Each Picatinny CLIK payload location shall be assigned a unique numerical identifier, starting at 1 and increasing sequentially. These identifiers will correspond to the platform payload channel identifiers as described in 4.2. Platforms with two or more CLIK connectors shall permanently mark (1 to *n*) is identifier within three inches of its associated connector with the format “CH1”, “CH2”, etc.

4.1.1 PICATINNY RAIL MARKINGS

Each singularly mounted Picatinny Rail will have the centerline identified and marked on both sides with a GO & No-GO range of locations so that the payload can be mounted and not adversely affect the flight control and operations of the UAS (see Figure 17). The UAS platform shall analyze where the best place and space of the rail(s).

The intent of this marking is to reduce, but not eliminate, burden of aircraft weight and balance calculations in an operational environment.

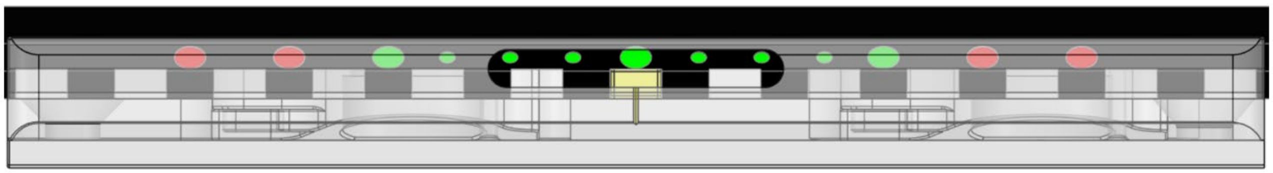


Figure 17 - Picatinny Rail w/centerline & defined GO & NO-GO markings, viewable thru Notional window design

When a singularly mounted Picatinny Rail is longer than the min 5.0” rail or continuously along the UAS mounting surface, it will have the centerline identified and marked with a range of locations that the payload may be mounted in “GO” range and additionally a “NO GO” where it would adversely affect the flight control and operations of the UAS (see Figure 18).

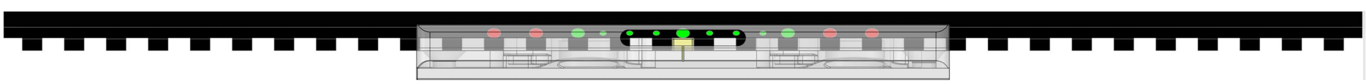


Figure 18 - Picatinny Rail longer than the min 5.0” w/centerline & defined GO & NO-GO markings viewable thru Notional window design

4.1.2 PICATINNY CLIK GRID

The platform requirements when hosting an array consist of units, CU, or CPU that create a repeatable grid of expected capabilities on the UAS.

A Platform may only have one physical interface but may allow a payload of 1x2 CU if the area under the platform can accommodate it. The CU and CPU are physical and electrical boundaries that are directly related to the capabilities offered by the payload.

System Integrators are responsible for verifying that the weight and dimensions of a CLIK Unit Payload, including those designated as WxL, are within the platform’s operational limits.

When mounting the rail below the UAS, allowances shall be made to accommodate wiring connected to the UAS as shown in Figure 6. No UAS protrusions may exist in this space.

For Picatinny CLIK array interfaces, MIL-STD-1913 accessory rails shall be used. The platform shall provide a rail of sufficient length, material, and attachment strength to support the platform maximum payload weight when subjected to 5G loading applied simultaneously at the platform alignment location.

Use MIL-STD-1913 as the standard for the mounting rail for payloads. The platform will specify the minimum length of the rail to be 5" via its attachment points and is dependent on the intended use of the rail whether it is longer than the minimum 5" or continuous up to the entire length of the platform. Any secondary attachment rails shall be placed on a 5" OC spacing with a clear attachment interface as seen in Figure 6. The program will specify the quantity of rails on a platform, depending on the intended use of the rails.

Payloads can be mounted on Picatinny rails. Single rails must be at least 5 inches long. If multiple rails are needed, they will run parallel to each other, with a center-to-center displacement of 5.0 inches (127mm). For additional rails along the long axis, they will be centered at 13.79 inches (350mm) and can be continuous as desired. The exact placement of these rails on the Unmanned Aircraft System (UAS) is determined by the payload's capabilities (lethal or non-lethal, including mechanisms) and the platform's maneuverability. For details and space claim on Picatinny CLIK ecosystems, see Figure 3.

Note: If the UAS does not have the nominal 8" ground clearance, a takeoff platform may be used to provide clearance. Picatinny CLIK Ecosystem Space Claim – 1 Unit

The platform shall conduct analysis to determine the best mounting configurations for multiple Picatinny CLIK payloads to support proper operations and capabilities (i.e. not hinder) of the UAS.

4.1.3 IDENTIFICATION MARKINGS

The Platform shall display a permanent identification plate with human readable text and Machine-Readable Information (MRI) in accordance with MIL-STD-130N. The identification plate shall include the following minimum information: manufacturer, part number, unique serial number, maximum allowed payload weight in pounds to one decimal as APW: XX.X, and maximum available payload current in Amperes to one decimal as MAC: XX.X (see 4.2.1.1).

4.2 POWER AND SIGNAL INTERFACE

Several requirements are put on both platforms and payloads to implement a Picatinny CLIK compliant system. To maximize flexibility for developers, the requirements are defined as functionalities where possible rather than physical implementations. The primary functional building blocks of a Picatinny CLIK platform implementation include a single Picatinny CLIK Platform Interface (PI) and one or more Picatinny CLIK Payload Channels (PC). Each Payload Channel provides the functionality required to support a corresponding payload. The functional architecture of a Picatinny CLIK system is shown in Figure 19.

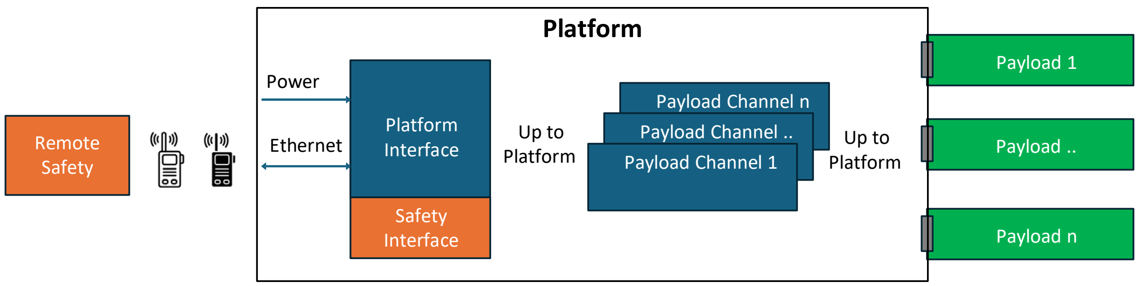


Figure 19 – Picatinny CLIK Functional Architecture

The backbone of Picatinny CLIK communication interfaces relies on an Internet Protocol bus via Ethernet. While the flexible architecture of Picatinny CLIK permits the Picatinny CLIK functions to be implemented in various ways, those functions will exist at distinct IP address and port pairs. The individual channels are identified in Table 11 along with references to the sections describing the specific requirements for each. Note that each channel would include its own IP interfaces for each function. In addition, a payload's IP interface configuration is limited to assigning a non-conflicting IP address(es). The ports used by the payload and the number of unique IP devices are left to the payload developer.

IP/Port Pair	Description	Reference
Platform Interface UDP	UDP interface to Platform Interface	4.2.1.2
Platform Safety UDP	UDP interface to Platform Safety Interface	4.2.2.1
Payload Channel [1..n] UDP	UDP interface to Payload Channel	4.2.3.3
Payload Channel [1..n] Serial	UDP to Serial converter for Payload Channel	4.2.3.4
Payload [1..n]	Optional IP interface(s) for connected Payload	3.2.5
Remote Safety UDP	UDP interface to Remote Safety	5.1

Table 11: Picatinny CLIK IP/Port Pairs

To better illustrate the hardware agnostic approach to the Picatinny CLIK network architecture two example implementations will be provided for a notional Picatinny CLIK platform supporting three payload channels. In this example each payload includes support for a single network device. In the example diagrams the following notation is used to represent the IP/Port pairs: Platform Interface (PI), Safety Interface (SI), Payload Channel N UDP Interface (CNI), Payload Channel N Serial Interface (CNS), Payload N Interface (PN). The respective IP address and subnet masks are represented using Classless Inter-Domain Routing (CIDR) notation, and the selected ephemeral port selected follows with a P prefix.

The notional implementation in Figure 20 uses an expanded architecture where several different devices (represented as a blue box) implement their respective interfaces. These devices could be a microcontroller or similar device. In this example the platform uses separate devices for the Platform Interface and Safety Interface. In addition, each channel uses a separate device supporting the payload channel UDP and serial interfaces. Each payload device incorporates a single network interface. In this architecture each device requires a unique IP address. Since a unique interface is the combination of both IP address and port, port numbers can be reused across different devices.

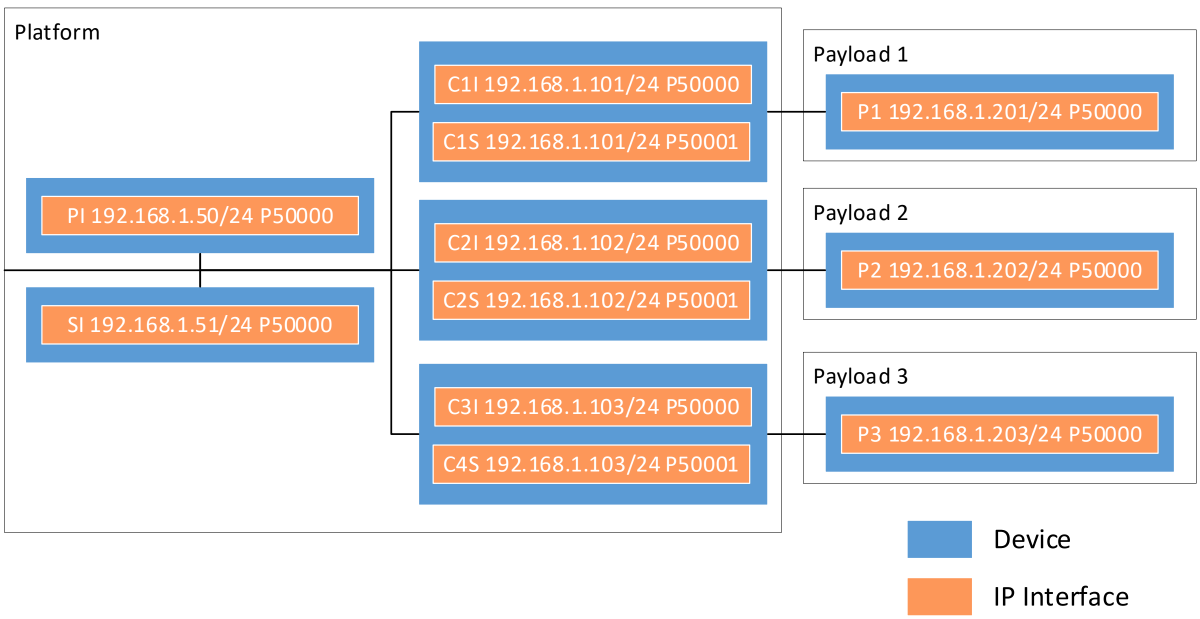


Figure 20 –Notional Network Devices (Expanded)

The notional implementation in Figure 21 uses a consolidated architecture where a single device in the platform implements all the interfaces necessary to support three payload channels. This approach is also valid. However, since the device implements a single IP address, each interface needs to be assigned to a unique port.

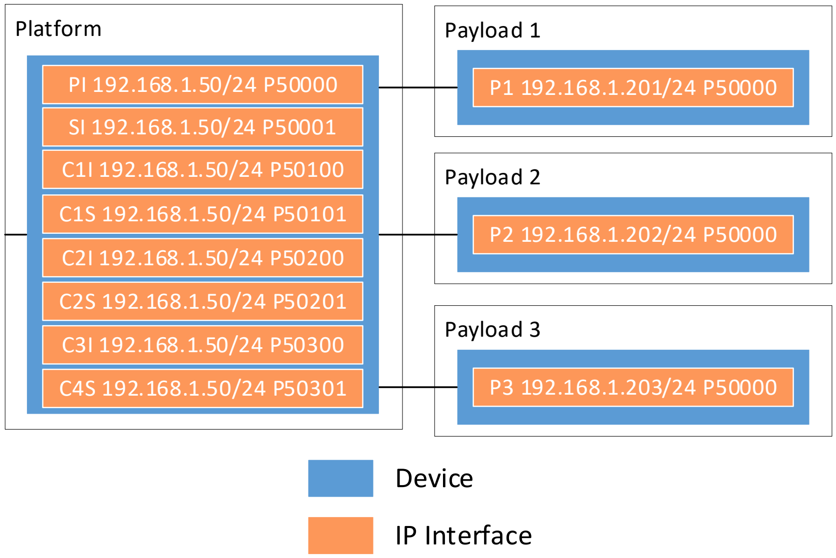


Figure 21 –Notional Network Devices (Condensed)

4.2.1 PLATFORM INTERFACE

The Platform Interface serves as the primary interface between the platform systems and the Picatinny CLIK functions. Raw power and an Ethernet interface are pulled from the platform. Regulated CLIK power, network interfaces, and safety signals are then provided to one or more Payload Channels as shown below in Figure 22.

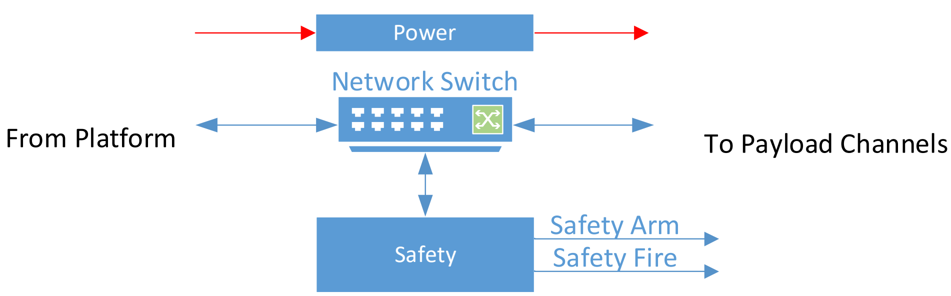


Figure 22 –Platform Interface Functions

4.2.1.1 POWER

The Platform Interface shall consume electrical power from the platform and convert it as needed to deliver the 28V power required by the Picatinny CLIK payloads. The developer shall determine the maximum available payload current (MAC) to be made available for use by payloads. Overcurrent protection shall be implemented such that failure of the Picatinny CLIK system cannot affect performance of the platform. The Platform Interface shall prevent back feeding of Picatinny CLIK power to the platform. The Platform Interface shall implement switching of the 28V power bus between the on and off state via the Set Platform Power message as described in 6.3.5. The Platform Interface shall default to CLIK power off when the system is initially powered on. Setting the CLIK power off only removes power provided to the payloads. The functionality provided by the Platform and Channel Interfaces shall persist regardless of CLIK power state.

4.2.1.2 ETHERNET

The Platform Interface shall connect to the platform's Ethernet network and implement an individual Ethernet interface for each supported payload channel and an Ethernet interface for the safety interface function. All interfaces shall be connected via full Layer 2 switch functionality. The Platform Interface shall support configuration of the IP addresses, subnet masks, and UDP ports for all platform network interfaces though how this is implemented is left to the platform developer. The IP addressing scheme is left to the system integrator.

The Platform Interface shall implement the UDP Protocol defined in 6.3. This interface provides the ability to request status/configuration data and issue commands. With the IP address and port of this interface a software application can identify the platform, determine the number of channels available, and collect the IP address and port pairs for each platform hosted interface.

4.2.2 SAFETY INTERFACE

Due to the safety critical nature of operating with lethal payloads, additional efforts are necessary to ensure that signals used for those functions are relayed accurately and timely. The diligence required to prove out these functions for the purpose of obtaining an approved safety release in accordance with MIL-STD-882E has driven the Picatinny CLIK architecture to break out this functionality to more readily support the integration of proven solutions. The resulting architecture utilizes a dedicated network interface controlling the output of two safety signals: Safety Enable and Safety Execute. These signals are distributed to payload channels and any connected payloads. Note that these two signals are global and shared throughout the platform and its payloads.

The safety critical aspects of the design are isolated to the essential required functions such that the stringent requirements for safety do not extend beyond where they are necessary. A platform is configured to communicate with a Remote Safety device (See Section 5) at its associated IP address and port. Once properly configured, the platform periodically transmits a request to the Remote Safety which then transmits a response containing the Safety

Enable and Safety Execute states currently set by the operator. If the message received satisfies all the requirements of a valid response the safety signal outputs are set to match the states received. Since all features required for safety are generated and verified at the platform, it is possible to treat the IP network as a black-channel with no additional safety requirements needed for the data transport medium.

As shown in Figure 23, safety communications are implemented using the safety architecture defined in Open Platform Communication's (OPC) Unified Architecture (UA) as defined in IEC 62541. The UA specification is composed of multiple parts with Part 15 describing a Safety Communication Layer (SCL) which allows safety critical devices to exchange safety related data. The SCL provides a means to operate in conjunction with UA specific services defined in other parts of the UA specification. The CLIK specification does not implement the UA communication protocols but rather sends the SCL messages over UDP as described in 4.2.2.1, 4.2.2.2.3, and 5.1.

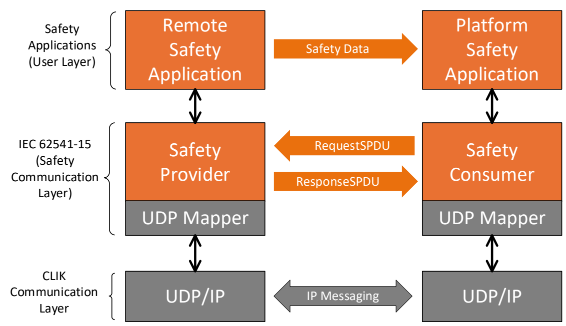


Figure 23 –Safety Communication Architecture

4.2.2.1 ETHERNET

The Safety Interface shall connect to the platform's Ethernet network. The Safety Interface shall support configuration of its IP address, subnet mask, and UDP ports, though how this is implemented is left to the platform developer. The IP addressing scheme is left to the system integrator.

The Safety Interface shall implement the UDP Protocol defined in 0. This interface provides the ability to request status/configuration data and issue commands.

The Safety Interface shall transmit RequestPDUs to the IP Address and Port configured by the Set Remote Safety Device message defined in 6.4.5 and listen for ResponsePDUs on the same port used for transmitting.

4.2.2.2 SAFETY CONSUMER

The Safety Interface of the Platform shall conform to the Safety Consumer requirements listed in Table 12 and defined in IEC 62541-15. Modifications listed in Table 12 shall be implemented in addition or in place of their respective requirement identifier.

Req. Id	Modifications
RQ4.1	The requirements defined in RQ4.1 shall only be met if the device is developed to a defined SIL.
RQ5.1	None
RQ5.2	None
RQ5.3	None
RQ5.4	None
RQ5.5	None
RQ5.6	None
RQ6.11	None
RQ6.14	The requirements defined in RQ6.14 shall adhere to the specific modifications to the SAPI fields defined in 4.2.2.2.1.
RQ6.15a/b	The requirements defined in RQ6.13 shall adhere to the specific modifications to the SPI fields defined in 4.2.2.2.2.
RQ6.16	None
RQ6.17	In meeting the requirements stated in RQ6.17, the Safety Consumer shall report Diagnostic messages using the protocol defined in 6.4.7.
RQ7.1	None
RQ7.2	None
RQ7.3	None
RQ7.4	None
RQ7.5	None
RQ7.6	None
RQ7.7	None
RQ7.8	None
RQ7.12	None
RQ7.14	None
RQ7.16	None
RQ7.17	In addition to values already defined in RQ7.17, a SafetyProviderLevel_ID value of 0x3A5F91C6 shall be used to represent a SafetyProviderLevel of 0.
RQ7.18	None
RQ7.19	None
RQ7.21	None
RQ7.22	None
RQ7.23	None
RQ7.24	None
RQ7.25	None

RQ7.26	None
RQ8.1	To support the calculation of SafetyConsumerTimeout the SafetyProvider shall provide the SafetyProviderDelay using the Get Safety Device Info message in 6.4.3. That value shall be provided to the SafetyConsumer using the Set Remote Safety Device message in 6.4.5.
RQ9.2a/b	None

Table 12: Safety Consumer IEC62541 Requirements

4.2.2.2.1 SAFETY APPLICATION PROGRAM INTERFACE VALUES

The Safety Application Program Interface (SAPI) defines an interface between the Platform Safety Application and the Safety Consumer which is used to exchange data during runtime. The Safety Interface shall implement the SAPI values shown in Table 13. Modifications listed in Table 13 shall be implemented in addition or in place of their respective SAPI term.

SAPI Term	Type	I/O	Modifications
SafetyData	Structure	O	Shall implement the data structure defined in 4.2.2.2.3
NonSafetyData	Structure	O	No non-safety data shall be included.
Enable	Boolean	I	None
FSV_Activated	Boolean	O	None
OperatorAckConsumer	Boolean	I	Shall be logically connected to OperatorAckProvider.
OperatorAckRequested	Boolean	O	None
OperatorAckProvider	Boolean	O	Shall be logically connected to OperatorAckConsumer.
TestModeActivated	Boolean	O	None
SafetyProviderID	UInt32	I	Shall initialize to “0” but be settable via 6.4.5
SafetyBaseID	GUID	I	Shall initialize to “0” but be settable via 6.4.5
SafetyConsumerID	UInt32	I	None

Table 13: Safety Consumer SAPI Values

4.2.2.2.2 SAFETY PARAMETER INTERFACE VALUES

The Safety Parameter Interface (SPI) defines an interface between the Platform Safety Application and the Safety Consumer which is used to exchange data used during commissioning. The Safety Interface shall implement the SPI values shown in Table 14. Modifications listed in Table 14 shall be implemented in addition or in place of their respective SPI term.

SAPI Term	Type	Modifications
SafetyProviderIDConfigured	UInt32	Shall be kept at value 0.

SafetyBaseIDConfigured	GUID	Shall be kept at value 0.
SafetyConsumerIDConfigured	UInt32	None
SafetyProviderLevel	Byte	Shall set in accordance with 4.2.2.4
SafetyStructureSignature	UInt32	None
SafetyStructureSignatureVersion	UInt16	None
SafetyStructureIdentifier	String	Shall use the string “CLIK”
SafetyOperatorAckNecessary	Boolean	None
SafetyErrorIntervalLimit	UInt16	Shall accept other values than those explicitly listed if not adhereing to a defined SIL.
SafetyClientImplemented	Boolean	Shall set to “0”
SafetyPubSubImplemented	Boolean	Shall set to “0”

Table 14: Safety Consumer SPI Values

4.2.2.2.3 SAFETY PROTOCOL DATA UNIT

The SafetyData transmitted by the SafetyProducer shall consist of two Boolean values representing the state of the Safety Enable and Safety Execute signals in accordance with IEC 62541-15 RQ6.11. Each Boolean shall also have a corresponding qualifier indicating the validity of their respective signal in accordance with IEC 62541-15 RQ6.16. There are four additional bytes reserved for future use. All reserved values shall be set to 0. The fields shall be represented via the eight data bytes shown in Figure 24. The Fail-Safe Substitute values (FSV) for the Safety Enable and Safety Execute signals shall be False (0x0).

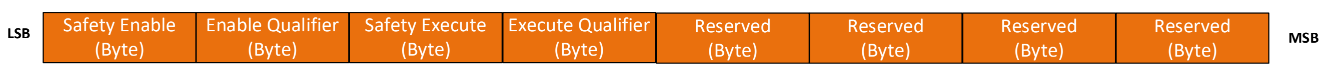


Figure 24 – Safety Data Content

For reference, a depiction of the components of the RequestSPDU and ResponseSPDU are depicted below in Figure 25 to include the UDP/IP headers. To comply with IEC 62541-15 RQ6.8 a Boolean dummy value of value 0 shall be used to represent the NonSafetyData portion of the Response SPDU.

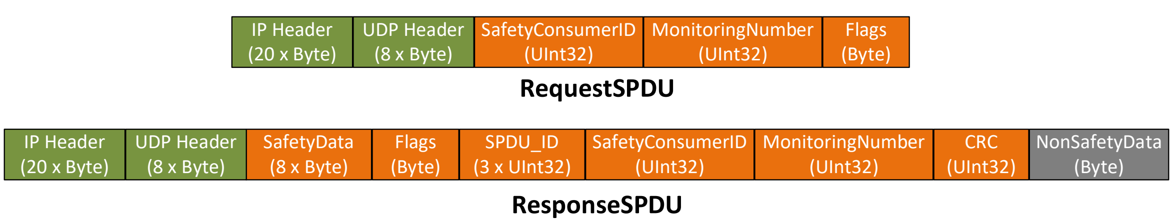


Figure 25 – Safety Protocol Data Unit Messages

To calculate the SafetyStructureSignature all eight SafetyData variables shall be treated as DataType Boolean with DataTypeEncodingId equal to ‘1’.

4.2.2.3 SAFETY SIGNALS

The Safety Interface shall generate the two safety signals and distribute them to its Payload Channels for delivery to their respective payloads. The current logical state of the safety signals is obtained from a chosen Remote Safety device in accordance with Section 5.

If the received SPDU data from the Remote Safety indicated that the Remote Safety Device has Activated its Test Mode, the Platform Safety Interface shall report the safety states received but shall set both electrical outputs to their fail safe substitute values as defined in 4.2.2.2.3.

Logic voltage output shall be 5V nominal with greater than or equal to 4.9V output for an active signal and less than or equal to .1V for an inactive signal. Each safety signal shall be capable of supplying current of no less than the number of payload channels multiplied by 10mA while maintaining the required voltage output. The Safety Interface shall provide an independent signal return path for the safety signals. The safety signals shall be shared by all Payload Channels and ultimately the Payloads themselves.

Once generated by the safety interface, the Safey Enable, Safety Execute, and Safety Return signals shall be electrically isolated from all other signals using isolation devices in accordance with International Electrotechnical Commission (IEC) 60747-5-5 or IEC 60747-17. This isolation requirement shall be enforced at the system level to include the platform and any connected payloads.

4.2.2.4 FUNCTIONAL SAFETY LEVEL

This document is limited to defining the requirements for safely communicating the state of the safety signals over an IP network. Additional requirements for designing electronic circuitry and software to meet a given standard of safety are outside the scope of this document. If the developer chooses to develop and certify the safety functions to meet a Safety Integrity Level (SIL) as defined in IEC 61508 then the rated SIL level, from level 1 to level 4, shall be reported in accordance with IEC 62541-15. For devices that are not SIL rated a value of 0 shall be reported in place of SIL level.

The intent is for safety devices to be implemented for the intended operations. Less stringent safety requirements can potentially make smaller and lower cost devices available in situations where safety risks are low or where a higher level of risk is tradable. Since the SIL level (or absence if SIL level) is communicated between the Remote Safety and Platform Safety Interface, the software and/or operator can restrict use or alert the operator to a mismatch. How this is handled is left to the system developer based on the specific needs of the intended application.

4.2.3 PAYLOAD CHANNELS

The Payload Channels implement the signals used by a Picatinny CLIK payload as defined in 3.2.3. For every Picatinny CLIK payload supported, an associated Payload Channel shall be implemented. Power, network interfaces, and safety signals are taken from the Platform Interface while the Payload Channel provides power, a network interface, serial interface, safety signals and a loopback signal to the channel's associated payload. This is depicted in Figure 26.

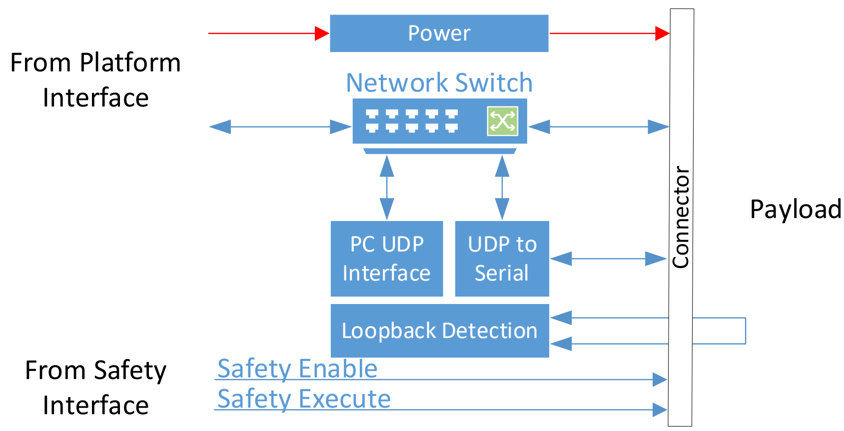


Figure 26 –Payload Channel Functions

4.2.3.1 PLATFORM CONNECTORS

Platforms shall integrate connectors to interface signals from the Platform Channels to the Payloads. Platforms shall use current available options shown in Table 15 or connectors with identical form, fit, and function as detailed below.

<table>
<tr><th>Series</th><th>Class</th><th>Shell Style</th><th>Material/Finish</th><th>Arrangement</th><th>Contacts</th><th>Keying</th></tr>
<tr><td>801</td><td rowspan="2">-009</td><td>-01</td><td rowspan="2">NF</td><td rowspan="4">9-19</td><td rowspan="4">S</td><td rowspan="4">A</td></tr>
<tr><td rowspan="2">2M801</td><td rowspan="2">-02</td></tr>
<tr><td rowspan="2">-010</td><td rowspan="2">ZNU</td></tr>
<tr><td>MD801</td><td>-07</td></tr>
</table>

Table 15: Approved Options for Platform Connectors

Picatinny CLIK Platform Connectors shall conform to the series, class, style, material/finish, arrangement, and keying defined in 3.2.1.

Picatinny CLIK Platform Connectors shall use Female (Socket) Contacts of Size #23.

4.2.3.2 POWER

Payload Channels shall consume electrical power from the Platform Interface and provide that power to their Picatinny CLIK payloads. Payload Channels shall implement overcurrent protection to ensure that a Picatinny CLIK payload cannot exceed the lesser of the MAC as defined in 0 or the MCC as defined in 3.2.4. Payload channels shall not provide power to a payload if the loopback is not present. A loopback signal shall be present continuously for no less than .5 seconds and no more than 1 second before electrical power is provided to the payload. A Platform shall not be damaged if a payload is disconnected while being actively powered.

For CLIK platforms of integration level 3 and above the following requirements shall be met. Payload Channels shall implement and report the power options represented in the Get Payload Channel Power Config message as described in 6.5.2. Payload Channels shall implement switching of the 28V payload power between the “On” and “Off” state via the Set Payload Channel Power message as described in 6.5.3. Payload channels shall default to power off when the system initially powers on. Payload Channels shall optionally implement overcurrent resetting via the Set Payload Channel Power message as described in 6.5.3 with an “On” command resetting the overcurrent device and restoring power. Payload Channels shall optionally implement configurable current limit setting via the Set Payload Channel Current Limit message as described in 6.5.4.

4.2.3.3 ETHERNET

Payload Channels shall connect to the Ethernet interface instance provided by the Platform Interface. All interfaces shall be connected via full Layer 2 switch functionality. Payload Channels shall implement the Ethernet interface defined in 0 for channel state information and configuration. Assignment of a Payload Channel’s IP address and ports are left to the system integrator.

4.2.3.4 SERIAL

Payload Channels shall implement a UDP to bi-directional serial converter using their provided network interface. Payload Channels shall listen for network datagrams on their configured port and once received immediately transmit the raw data to the payload’s RS232 serial interface. All serial data received from the payload’s RS232 serial interface shall be discarded until communication has been initiated via receipt of a network datagram. Once a datagram has been received, data received from the payload’s RS232 Rx pin shall be packetized and returned to the IP address and source port of the most recent datagram on the indicated network port for serial transfer.

The RS232 serial interface implemented for each payload channel shall support no less than 115,200 bps data rate. Each RS232 interface channel shall implement independent functional isolation from other electrical signals. RS232 signals shall be referenced to the channel’s isolated RS232 Return signal. This isolation prevents ground loops caused by the RS232 channel’s Return line serving as an additional return path for the CLIK Power Bus provided to the payload. Functional isolation shall be no less than +/-50Vrms continuous and +/-100Vrms transient for 60 seconds.

4.2.3.5 SAFETY SIGNALS

Payload Channels shall accept the safety signals provided by the Safety Interface and pass them on to their respective payloads. Safety signals shall be measured by the payload channels for state reporting to indicate the state of the safety signals being fed to each payload. The measurement of these signals shall follow the requirements specified in 3.2.8. Measurement of these signals shall occur as close to the payload as is technically feasible. For example, where signals are split the measurement should occur after the split, closer to the payload.

4.2.3.6 LOOPBACK

Payload Channels shall implement a loopback circuit for the detection of a connected payload. Output voltage from the signal shall be less than or equal to the power voltage and when shorted to common shall consume less than 1mA of current. An example of a suitable circuit would be a 10kOhm resistor pulled up to 5V connected to a digital input which gets pulled low when a payload and its loopback jumper are connected.

4.2.4 GROUNDING

The Picatinny CLIK platform grounding scheme shall be carefully designed to eliminate ground loops. The CLIK platform shall maintain isolation between its Chassis signal and all other electrical signals. The platform host power ground should have a single connection path to any CLIK ground references. The decision to bond or isolate the platform chassis to or from other ground references outside the CLIK subsystem is left to the discretion of the platform developer.

4.3 PLATFORM TO PAYLOAD CABLING

Insulated Cables shall be utilized by Platforms to connect to Payloads and distribute Power and Signals between the Platform Channels and the Payloads as shown in Figure 27. Platform and Payload Receptacle Connectors (specified in Sections 3.2.1 & 4.2.3.1) are denoted with ‘J’ designation. Cable Plug Connectors are denoted with ‘P’ designation.

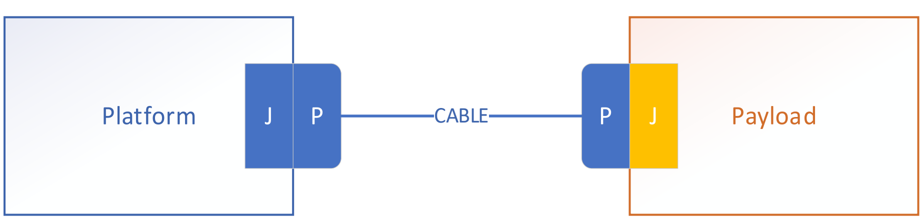


Figure 27: Platform to Payload Cables

4.3.1 CABLE CONNECTORS

Cables shall integrate connectors using available options shown in Table 16 or connectors with identical form, fit, and function as detailed below.

<table>
<tr><th>Series</th><th>Class</th><th>Shell Style</th><th>Material/Finish</th><th>Arrangement</th><th>Contacts</th><th>Keying</th></tr>
<tr><td>801</td><td rowspan="2">-007</td><td rowspan="2">-16</td><td rowspan="2">NF</td><td rowspan="4">9-19</td><td rowspan="4">*See Below</td><td rowspan="4">A</td></tr>
<tr><td rowspan="2">2M801</td></tr>
<tr><td rowspan="2">-008</td><td rowspan="2">-26</td><td rowspan="2">ZNU</td></tr>
<tr><td>MD801</td></tr>
</table>

Table 16 Approved Options for Cable Connectors

Picatinny CLIK Cable Connectors shall conform to the series, material/finish, arrangement, and keying defined in 3.2.1.

Picatinny CLIK Cable Connectors shall be of Plug Class Series options shown in Table 17. The banding platform option incorporates features to support the attachment of shielding and/or shrink boots while the rear accessory thread option requires a separate backshell (i.e. straight, 45-degree, 90-degree). Examples of both are shown in Figure 28.

Figure 28: Plug Connector Series Options

Series	Description
-007	Plug with Banding Platform
-008	Plug with Rear Accessory Thread

Table 17: Approved Connector Series

The approved manufacturers provide options for Shell Style. Cables can use the Shell Style Options in Table 18 that best suits their application.

Shell Style	Description
-16	Anti-Decoupling Spring
-26	Self-Locking Ratchet Mechanism

Table 18: Shell Style Options

The Picatinny CLIK Cable Connector that mates with the Payload shall use Option ‘S’ for Female (Socket) Contacts of Size #23.

The Picatinny CLIK Cable Connector that mates with the Platform shall use Option ‘P’ for Male (Pin) Contacts of Size #23.

4.3.2 POWER

Conductor Wiring for Power Signals shall be of Size 22 American Wire Gauge (AWG).

4.3.3 ETHERNET

Acceptable Sizing for Ethernet Signal Conductor Wiring shall be 22 to 28AWG in Twisted Pairs for Rx & Tx Signals.

4.3.4 SERIAL

Acceptable Sizing for Serial Signal Conductor Wiring shall be 22 to 28AWG.

4.3.5 SAFETY SIGNALS

Acceptable Sizing for Safety Signal Conductor Wiring shall be 22 to 28AWG.

4.3.6 LOOPBACK

Acceptable Sizing for Loopback Signal Conductor Wiring shall be 22 to 28AWG.

5 REMOTE SAFETY REQUIREMENTS

The Remote Safety is responsible for capturing inputs from the operator and transmitting their input states to platforms for generation of the Safety Enable and Safety Execute signals that are distributed to the payloads. The Remote Safety functionality will typically be co-located with the CLIK payload operator. The form of the operator inputs is not defined in this document. The physical form of these inputs and how they are named are left to the developer to best suit the operator and the application. For example, for one application the Safety Enable signal may correlate to a toggle switch input labeled “Arm” and the Safety Execute signal may be paired with a momentary trigger switch labeled “Fire”. Other applications may be better addressed with different styles of switches and different nomenclature. In some applications it may be possible to rely on inputs captured within a software interface. See 4.2.2 for additional details on how the platform and remote safety interact with each other.

A Remote Safety device shall implement a means to set the device into Test Mode as defined in 5.2.1 via the EnableTestMode term. A Remote Safety device shall implement a means to indicate to the operator if the device is in its normal operating mode or its test mode. How these requirements are implemented are left to the developer and any additional safety requirements. This could be a physical switch and indicator or via a graphical user interface.

A Remote Safety device shall implement a means to indicate that an Operator Acknowledge has been requested by the platform safety interface as defined in 5.2.1 via the OperatorAckRequested term. A Remote Safety device shall implement a means for an operator to provide the Operator Acknowledge as defined in 5.2.1 via the OperatorAckProvider term. How these requirements are implemented are left to the developer and any additional safety requirements. This could be a physical switch and indicator or via a graphical user interface.

5.1 ETHERNET

The Remote Safety shall connect to the communication Ethernet network associated with the Platforms to be paired with. This would typically be accomplished via a network radio interface though the specifics of how this is done are left to the developer/integrator. The Remote Safety shall support configuration of its IP address, subnet mask, and UDP ports, though how this is implemented is left to the developer. The IP addressing scheme is left to the system integrator.

The Remote Safety shall implement the UDP Protocol defined in 0. This interface provides the ability to request status/configuration data and issue commands.

The Remote Safety shall listen for RequestPDUs on its configured IP Address and UDP Port as reported by its Get Remote Safety Device message defined in 6.4.4. The Remote Safety shall generate ResponsePDUs as defined in 5.2 and transmit the response to the source address and source port defined in the received UDP/IP packet.

5.2 SAFETY PRODUCER

The Remote Safety shall conform to the Safety Producer requirements listed in Table 19 and defined in IEC 62541-15. Modifications listed in Table 19 shall be implemented in addition or in place of their respective requirement identifier.

Req. Id	Modifications
RQ4.1	The requirements defined in RQ4.1 shall only be met if the device is developed to a defined SIL.
RQ5.1	None
RQ5.2	None
RQ5.3	None
RQ5.4	None
RQ5.5	None
RQ5.6	None
RQ6.11	None
RQ6.12	The requirements defined in RQ6.12 shall adhere to the specific modifications to the SAPI fields defined in 5.2.1
RQ6.13a/b	The requirements defined in RQ6.13 shall adhere to the specific modifications to the SPI fields defined in 5.2.2
RQ6.16	None
RQ7.1	None
RQ7.2	None
RQ7.3	None
RQ7.4	None
RQ7.5	None
RQ7.6	None
RQ7.7	None
RQ7.8	None
RQ7.11	A Safety Consumer may request an identical RequestSPDU multiple times in a row. The Safety Provider shall respond to all requests while providing the most recently acquired safety data.
RQ7.13	None
RQ7.15	None
RQ7.16	None
RQ7.17	In addition to values already defined in RQ7.17, a SafetyProviderLevel_ID value of 0x3A5F91C6 shall be used to represent a SafetyProviderLevel of 0.
RQ7.18	None
RQ7.19	None

RQ7.21	None
RQ7.22	None
RQ7.23	None
RQ7.24	None
RQ8.1	To support the calculation of SafetyConsumerTimeout the SafetyProvider shall provide the SafetyProviderDelay using the Get Safety Device Info message in 6.4.3. That value shall be provided to the SafetyConsumer using the Set Remote Safety Device message in 6.4.5.
RQ9.1	None

Table 19: Safety Producer IEC62541 Requirements

5.2.1 SAFETY APPLICATION PROGRAM INTERFACE VALUES

The Safety Application Program Interface (SAPI) defines an interface between the Remote Safety Application and the Safety Producer which is used to exchange data during runtime. The Remote Safety shall implement the SAPI values shown in Table 20. Modifications listed in Table 20 shall be implemented in addition or in place of their respective SAPI term.

SAPI Term	Type	I/O	Modifications
SafetyData	Structure	I	Shall implement the data structure defined in 4.2.2.2.3
NonSafetyData	Structure	I	No non-safety data shall be included.
EnableTestMode	Boolean	I	None
OperatorAckProvider	Boolean	I	None
OperatorAckRequested	Boolean	O	None
ActivateFSV	Boolean	I	None
SafetyConsumerID	UInt32	O	None
MonitoringNumber	UInt32	O	None
SafetyProviderID	UInt32	I	None
SafetyBaseID	GUID	I	Shall always be set to “0”

Table 20: Safety Producer SAPI Values

5.2.2 SAFETY PARAMETER INTERFACE VALUES

The Safety Parameter Interface (SPI) defines an interface between the Remote Safety Application and the Safety Producer which is used to exchange data used during commissioning. The Remote Safety shall implement the SPI values shown in Table 21. Modifications listed in Table 21 shall be implemented in addition or in place of their respective SPI term.

SAPI Term	Type	Modifications
SafetyProviderIDConfigured	UInt32	Shall be set by developer IAW Section 9.1.1 of IEC 62541-15.
SafetyBaseIDConfigured	GUID	Shall be set by developer IAW Section 9.1.1 of IEC 62541-15.
SafetyProviderLevel	Byte	Shall set in accordance with 4.2.2.4
SafetyStructureSignature	UInt32	None
SafetyStructureSignatureVersion	UInt16	None
SafetyStructureIdentifier	String	Shall use the string "CLIK"
SafetyProviderDelay	UInt32	None
SafetyClientImplemented	Boolean	Shall set to "0"
SafetyPubSubImplemented	Boolean	Shall set to "0"

Table 21: Safety Producer SPI Values

5.3 FUNCTIONAL SAFETY LEVEL

The developer shall develop the Remote Safety to meet a Functional Safety Level in accordance with the requirements in 4.2.2.4.

6 COMMUNICATION PROTOCOL REQUIREMENTS

6.1 COMMON MESSAGE PROTOCOL FORMAT

A common protocol shall be used for all Picatinny CLIK defined messages. Table 22 defines the contents of a Picatinny CLIK message.

Byte	Data Type	Description
1	Header 1	Fixed header value of 0xAA hexadecimal.
2	Header 2	Fixed header value of 0x55 hexadecimal.
3	Message Identifier MSB	Unique value that identifies message
4	Message Identifier LSB	
5	Sync Code	Client provided value that is repeated back in response.
6	# Data Bytes	Total number of data bytes (n)
7	Data 1	First data byte
..	..	..
N - 2	Data n	Last data byte
N-1	CRC MSB	Cyclic redundancy check of bytes 1 through N – 2.
N	CRC LSB	

Table 22: Message Structure

The first two header bytes shall indicate the start of a new message with a 0xAA byte followed by a 0x55 byte (hexadecimal). The third and fourth bytes shall form a unique 16-bit message identifier indicating the contents of the message data with the most significant byte followed by the least significant byte. The fifth byte shall be reserved for a synchronization byte which the receiver shall duplicate in its response message. The sixth byte shall indicate the total number of data bytes included in the message. If one or more data bytes are indicated, then that data will start with byte seven.

The final two bytes shall contain a CRC of all preceding bytes. The CRC shall be computed using the CRC-16/CCITT-FALSE algorithm with the following parameters:

- Polynomial: 0x1021
- Initial value: 0xFFFF
- No input or output reflection
- No final XOR

Upon detection of a valid header the receiver shall allow 500ms for the complete message to be received. If the time limit is exceeded the incoming buffer shall be discarded and the receiver shall return to waiting for a valid header.

Message Identifiers in the range of 0x0000 to 0x7FFF shall be reserved for the Picatinny CLIK interface, with all remaining identifiers from 0x8000 to 0xFFFF available for application specific use. Message Identifiers are unique to a specific protocol and reuse of Message Identifiers is permitted on different channels.

The Common Picatinny CLIK Protocol adheres to a client/server model with each role fixed. Messages from a client shall always be responded to by the server. Accepted messages shall respond using the same Message Identifier. The response can include data specific to the received message or a more general acknowledge response that acknowledges the message or provides a meaningful error code indicating why the message was rejected. Messages that request data without any additional parameters shall not require any data as part of its message.

Messages returned in response to a request for data shall contain the specified data or a single byte with an error code indicating why the request was invalid. Data responses messages shall always include more than one byte to differentiate error responses from data responses. Messages returned in response to a command shall always return a single byte indicating the result of the command. As with the message identifiers, response result codes in the range of 0x00 to 0x7F shall be reserved for the Picatinny CLIK interface, with all remaining codes available for application specific use. Table 23 lists the codes currently supported by the Picatinny CLIK interface.

Code	Description
0x00	Command was processed successfully.
0x01	Bad checksum
0x02	Invalid Command
0x03	Invalid Data Length
0x04	Invalid Command Parameter
0x05	Not valid in current state

Table 23: Response Result Codes

6.2 PAYLOAD SERIAL PROTOCOL

The Payload Serial Protocol defines the interface implemented by the payload on its RS232 serial port. For this protocol the Payload is defined as the Server and shall only be transmitted in response to a received message. The following sections define the Picatinny CLIK defined message set that the Payload shall accept and the possible responses. This message set can be expanded as permitted in 6.1. Table 24 identifies the payload messages defined by CLIK to include their specific message identifiers and whether or not their implementation is required.

Msg Id	Description	Data Description	Required
0x0000	Get Payload Status	6.2.1	X
0x0001	Get Payload Identifier	6.2.2	X
0x0002	Get Payload System Config	6.2.3	X
0x0003	Get Function State Identifiers	6.2.4	X
0x0004	Set Function State	6.2.5	X
0x0005	Get Payload Network Config	6.2.6	
0x0006	Set Payload Network Config	6.2.7	
0x0007	Set Payload Network State	6.2.8	

Table 24: Payload Serial Message Identifiers

6.2.1 GET PAYLOAD STATUS

A payload shall respond to a Get Payload Status message with the data shown in Table 25. The number of available network interfaces can be used to enumerate their respective address information using the Get Payload Network Config message. Each function shall be allocated a data byte to represent its state. For example, a payload with 9 functions would require 9 additional bytes after the Number of Functions byte. Function states shall be assigned starting with Function 1 assigned to data byte 5 and increasing upwards.

Data	Data Type	Description
1	Status Bit Fields	Bitwise state data. See Table 26
2	Available Net Interfaces	Total number of network interfaces
3	Enabled Net Interface	Number of network interfaces enabled
4	Number of Functions	Number of functions implemented by the payload
..	Function states	0 or more bytes indicating function states as identified in Table 10.

Table 25: Payload Status Response Data Structure

Bit #	Name	Description	
0	Health	1=Functional	0=Not functional
1	Safety Enable state	1=Activated	0=Not activated
2	Safety Execute state	1=Activated	0=Not activated
3	Functions Available	1=1 or More	0=None
4	Functions Activated	1=1 or More	0=None
5	Functions Armed	1=1 or More	0=None
6	Reserved		
7	Reserved		

Table 26: Payload Status Bit Fields

6.2.2 GET PAYLOAD IDENTIFIER

A payload shall respond to a Get Payload Identifier message with the data shown in Table 27. The descriptive identifier shall be used by the payload to identify itself and shall be represented using no more than 32 characters represented using printable seven-bit American Standard Code for Information Interchange (ASCII) codes terminated by a null (zero) byte. The unique identifier field shall optionally be used by the payload to uniquely identify individual payload units (i.e. serial number) as an unsigned integer. The version field shall optionally be used by the payload to indicate a particular version or configuration of the payload. The version shall be represented using no more than 16 characters represented using printable seven-bit ASCII codes terminated by a null (zero) byte. A value of 0 shall be used for the unique identifier (UID) and Version fields if they are not used.

Data	Data Type	Description
1	UID MSB	Unique identifier most significant byte.
2	UID LSB	Unique identifier least significant byte.
..	Identifier	ASCII text identifying payload
..	Version	Configuration version of the payload

Table 27: Payload Identifier Response Data Structure

6.2.3 GET PAYLOAD SYSTEM CONFIG

A payload shall respond to a Get Payload System Config message with the data shown in Table 28.

Data	Data Type	Description
1	MPC	Maximum Payload Current in 10ths of Amps.
2	MPW MSB	Unsigned 16-bit integer representing maximum payload weight in tenths of pounds.
3	MPW LSB	
4	Safety Signals Monitored	1=Yes 0=No

Table 28: Payload System Config Response Data Structure

6.2.4 GET FUNCTION STATE IDENTIFIERS

A payload shall respond to a Get Function State Identifiers message with the data shown in Table 29. Each state (as defined in Table 10) supported by the payload shall be returned as a combination of the function state identifier followed by the function state name. Function state names shall be no longer than 16 characters and represented using printable seven-bit American Standard Code for Information Interchange (ASCII) codes terminated by a null (zero) byte. Unused states shall not be returned.

Data	Data Type	Description
1	Number of States	Number of states used by the payload
2	State Identifier 1	First state identifier used
..	State Name 1	ASCII text identifying first state
..	State Identifier 2	Next sequential state identifier
..	State Name 2	ASCII text identifying second state
..	State Identifier X	Final state identifier
..	State Name X	ASCII text identifying final state

Table 29: Function Identifier Response Data Structure

6.2.5 SET FUNCTION STATE

A payload shall accept a Set Function state message with the data shown in Table 30. If the message is valid the Payload shall set the indicated function channel to the commanded state and respond with a single data byte set to 0x00 for success. If rejected, a byte containing the error code shall be returned.

Data	Data Type	Description
1	Function Channel	Function channel to set.
2	Function state	1=Activate 0=De-Activate

Table 30: Set Payload Function State Receive Data Structure

6.2.6 GET PAYLOAD NETWORK CONFIG

A payload shall accept a Get Payload Network Config message formatted with the data shown in Table 31

Data	Data Type	Description
1	Selected Interface	Network interface to return. 1-255

Table 31: Get Payload Network Config

A payload shall respond to a Get Payload Network Config message with the data shown in Table 32. If the interface does not provide an IP interface it shall respond with a value of 0.0.0.0/0.

Data	Data Type	Description
1	Requested Interface	Interface requested
2	Interface state	1=Enabled 0=Disabled
3	IP Address Octet 1	NNN.NNN.NNN.NNN
4	IP Address Octet 2	NNN.NNN.NNN.NNN
5	IP Address Octet 3	NNN.NNN.NNN.NNN
6	IP Address Octet 4	NNN.NNN.NNN.NNN
7	Subnet Bits	Integer value indicating number of subnet bits (CIDR Notation)

Table 32: Payload Get Network Config Response Data Structure

6.2.7 SET PAYLOAD NETWORK CONFIG

A payload shall optionally accept a Set Payload Network Config message with the data shown in Table 33. If the message is valid the Payload shall change its IP address and subnet to the commanded values and respond with a single data byte set to 0x00 for success. If rejected, a byte containing the error code shall be returned.

Data	Data Type	Description
1	Selected Interface	Interface to configure
2	Network Interface state	1=Enabled 0=Disabled
3	IP Address Octet 1	NNN.NNN.NNN.NNN
4	IP Address Octet 2	NNN.NNN.NNN.NNN
5	IP Address Octet 3	NNN.NNN.NNN.NNN
6	IP Address Octet 4	NNN.NNN.NNN.NNN
7	Subnet Bits	Integer value indicating number of subnet bits (CIDR Notation)

Table 33: Set Payload Network Config Receive Data Structure

6.2.8 SET PAYLOAD NETWORK STATE

A payload shall optionally accept a Set Payload Network state message with the data shown in Table 34. If the message is valid the Payload shall change the commanded network interface to the commanded state and respond with a single data byte set to 0x00 for success. If rejected, a byte containing the error code shall be returned.

Data	Data Type	Description
1	Selected Interface	Interface to set. 0 to set all.
2	Network Interface state	1=Enabled 0=Disabled

Table 34: Set Payload Network State Receive Data Structure

6.3 PLATFORM INTERFACE UDP PROTOCOL

The Platform Interface UDP Protocol defines the interface implemented by the Picatinny CLIK Platform Interface. For this protocol the Platform Interface is defined as the Server and shall only be transmitted in response to a received message. The following sections define the Picatinny CLIK defined message set that the Platform Interface shall accept and the possible responses. This message set can be expanded as permitted in 6.1. Table 35 identifies the payload messages defined by CLIK to include their specific message identifiers and whether or not their implementation is required.

Msg Id	Description	Data Description	Required
0x0000	Get Platform Interface Status	6.3.1	X
0x0001	Get Platform Identifier	6.3.2	X
0x0002	Get Platform System Config	6.3.3	X
0x0003	Get Platform Network Config	6.3.4	X
0x0004	Set Platform Power	6.3.5	X

Table 35: Platform Interface UDP Protocol

6.3.1 GET PLATFORM INTERFACE STATUS

A Platform Interface shall respond to a Get Platform Interface Status message with the data shown in Table 36. An unsupported value shall be represented by the upper limits of the field (i.e. 255, 65535).

Data	Data Type	Description
1	Payloads Supported	Number of Payload Channels supported by the platform
2	Payloads Connected	Number of Payloads currently connected
3	Status Bit Fields	Bitwise state data. See Table 37.
4	Platform Voltage In MSB	Unsigned 16-bit integer representing raw platform input voltage in millivolts.
5	Platform Voltage In LSB	
6	Platform Current Draw MSB	Unsigned 16-bit integer representing platform current draw in milliamps.
7	Platform Current Draw LSB	
8	Picatinny CLIK Power Voltage Out MSB	Unsigned 16-bit integer representing Picatinny CLIK Power Bus voltage in millivolts.
9	Picatinny CLIK Power Voltage Out LSB	
10	Picatinny CLIK Power Current Draw MSB	Unsigned 16-bit integer representing Picatinny CLIK Power Bus current consumption in milliamps.
11	Picatinny CLIK Power Current Draw LSB	

Table 36: Platform Interface Status Response Data Structure

Bit #	Name	Description
0	Overcurrent Tripped	1=Yes 0=No
1	Picatinny CLIK Power Bus state	1=On 0=Off
2	Safety Enable state	1=Activated 0=Not activated
3	Safety Execute state	1=Activated 0=Not activated
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	

Table 37: Platform Status Bit Fields

6.3.2 GET PLATFORM IDENTIFIER

A platform shall respond to a Get Platform Identifier message with the data shown in Table 38. The descriptive identifier shall be used by the platform to identify itself and shall be represented using no more than 32 characters represented using printable seven-bit ASCII codes terminated by a null (zero) byte. The unique identifier field shall optionally be used by the platform to uniquely identify individual platforms as an unsigned integer. The version field shall optionally be used by the platform to indicate a particular version or configuration of the platform. The version shall be represented using no more than 16 characters represented using printable seven-bit ASCII codes terminated by a null (zero) byte. A value of 0 shall be used for the UID and Version fields if they are not used.

Data	Data Type	Description
1	UID MSB	Unique identifier most significant byte.
2	UID LSB	Unique identifier least significant byte.
..	Identifier	ASCII text identifying platform
..	Version	Configuration version of the platform

Table 38: Get Platform Identifier

6.3.3 GET PLATFORM SYSTEM CONFIG

A Platform Interface shall respond to a Get Platform System Config message with the data shown in Table 39.

Data	Data Type	Description
1	MAC	Maximum Available Current in 250mA units.
2	APW MSB	Unsigned 16-bit integer representing total Allowed Payload Weight in tenths of pounds.
3	APW LSB	

Table 39: Platform System Config Response Data Structure

6.3.4 GET PLATFORM NETWORK CONFIG

The Platform Interface shall accept a Get Platform Network Config message formatted with the data shown in Table 40.

Data	Data Type	Description
1	Selected Interface	0: Platform Interface UDP Channel 255: Safety Interface Channel N: Payload Channel N UDP Channel N+127: Payload Channel N Serial Converter Channel

Table 40: Get Platform Network Config

The Platform Interface shall respond to a Get Platform Network Config message with the data shown in Table 41 for the indicated interface. If the interface does not provide an IP interface it shall respond with a value of 0.0.0.0/0.

Data	Data Type	Description
1	Requested Interface	See Table 40
2	IP Address Octet 1	NNN.NNN.NNN.NNN
3	IP Address Octet 2	NNN.NNN.NNN.NNN
4	IP Address Octet 3	NNN.NNN.NNN.NNN
5	IP Address Octet 4	NNN.NNN.NNN.NNN
6	Subnet Bits	Integer value indicating number of subnet bits (CIDR Notation)
7	UDP Port MSB	UDP port associated with indicated interface
8	UDP Post LSB	

Table 41: Payload Get Network Config Response Data Structure

6.3.5 SET PLATFORM POWER

The Platform Interface shall accept a Set Platform Power message with the data shown in Table 42. If message is valid the Platform Interface shall set the Picatinny CLIK power bus to the commanded state. If rejected, a byte containing the error code shall be returned. If an overcurrent event has occurred a command to turn the bus state on shall reset the overcurrent device.

Data	Data Type	Description
1	Picatinny CLIK Power Bus state	1=On 0=Off

Table 42: Set Platform Power Receive Data Structure

6.4 SAFETY INTERFACE UDP PROTOCOLT

The Safety Interface Protocol defines the interface implemented by the Platform Safety Interface and the Remote Safety on their Ethernet interfaces. In both cases the aforementioned devices are defined as the Server and shall only transmit in response to a received message. The following sections define the Picatinny CLIK defined message set that the devices shall accept and the possible responses. This message set can be expanded as permitted in 6.1. Table 43 identifies the safety messages defined by CLIK to include their specific message identifiers and whether or not their implementation is required.

Msg Id	Description	Required	
		Platform	Remote
0x0000	Get Consumer Safety Status	X	
0x0001	Get Provider Safety Status		X
0x0002	Get Safety Device Info	X	X
0x0003	Get Remote Safety Device	X	X
0x0004	Set Remote Safety Device	X	
0x0005	Set Safety Consumer Enable	X	
0X0006	Get Safety Diagnostic Messages	X	

Table 43: Safety Message Identifiers

6.4.1 GET CONSUMER SAFETY STATUS

A Platform Safety Interface device shall respond to a Get Consumer Safety Status message with the data shown in Table 44. If rejected, a single byte containing the error code shall be returned.

Data	Data Type	Description
1	Device Status Bit Fields	Bitwise state data. See Table 26.
2	Device State Bit Fields	Bitwise state data. See Table 46.
3	Enabled	1=Yes 0=No

Table 44: Safety Consumer Status Data Structure

Values shown in Table 45 represent device status to include overall health and the state of the Safety Data as defined in 4.2.2.2.3.

Bit #	Name	Description
0	Health	1=Functional 0=Not functional
1	Safety Enable state	1=Activated 0=Not activated
2	Enable Qualifier state	1=Valid 0=Not valid
3	Safety Execute state	1=Activated 0=Not activated
4	Execute Qualifier state	1=Valid 0=Not valid
5	Reserved	
6	Reserved	
7	Reserved	

Table 45: Safety Device Status Bit Fields

Values shown in Table 46 represent states of the Producer and Consumer SAPI interfaces as defined in 5.2.1 and 4.2.2.2.1 respectively. FSV represents ActivateFSV for SafetyProviders and FSV_Activate for SafetyConsumers. Test Mode represents EnableTestMode for SafetyProviders and TestModeActivated for SafetyConsumers.

Bit #	Name	Description	
0	OperatorAckRequested	1=Requested	0=Not Requested
1	OperatorAckProvider	1=Requested	0=Not Requested
2	FSV	1=Activated	0=Not activated
3	Test Mode	1=Activated	0=Not activated
4	Reserved		
5	Reserved		
6	Reserved		
7	Reserved		

Table 46: Safety Device State Bit Fields

6.4.2 GET PROVIDER SAFETY STATUS

A Remote Safety device shall respond to a Get Provider Safety Status message with the data shown in Table 47. If rejected, a single byte containing the error code shall be returned.

Data	Data Type	Description
1	Device Status Bit Fields	Bitwise state data. See Table 26.
2	Device State Bit Fields	Bitwise state data. See Table 46.

Table 47: Safety Provider Status Data Structure

6.4.3 GET SAFETY DEVICE INFO

A safety device shall respond to a Get Safety Device Info message with the data shown in Table 48. The descriptive identifier shall be used by the device to identify itself and shall be represented using no more than 32 characters represented using printable seven-bit American Standard Code for Information Interchange (ASCII) codes terminated by a null (zero) byte. The unique identifier field shall optionally be used by the device to uniquely identify individual devices (i.e. serial number) as an unsigned integer. The version field shall optionally be used by the device to indicate a particular version or configuration of the device. The version shall be represented using no more than 16 characters represented using printable seven-bit ASCII codes terminated by a null (zero) byte. A value of 0 shall be used for the unique identifier (UID) and Version fields if they are not used. If rejected, a single byte containing the error code shall be returned.

Data	Data Type	Description
1	UID MSB	Unique identifier most significant byte.
2	UID LSB	Unique identifier least significant byte.
..	Identifier	ASCII text identifying safety device
..	Version	Configuration version of the safety device

Table 48: Safety Device Info Data Structure

6.4.4 GET REMOTE SAFETY DEVICE

A safety device shall respond to a Get Remote Safety Device message with the data shown in Table 49. This data represents the stored values of the device itself in the case of a Remote Safety, or the values to send requests to in the case of a Platform Safety Interface (as set via the Set Remote Safety Device message). If rejected, a single byte containing the error code shall be returned.

Data	Data Type	Description
1	IP Address Octet 1	NNN.NNN.NNN.NNN
2	IP Address Octet 2	NNN.NNN.NNN.NNN
3	IP Address Octet 3	NNN.NNN.NNN.NNN
4	IP Address Octet 4	NNN.NNN.NNN.NNN
5	UDP Port MSB	UDP port associated with Remote Safety SPDU interface
6	UDP Post LSB	
7	SafetyProviderID B4	Unsigned 32-bit integer representing the SafetyProviderID of the Remote Safety Device to connect to. Note that B0 indicates the least significant byte.
8	SafetyProviderID B2	
9	SafetyProviderID B1	
10	SafetyProviderID B0	
11	SafetyBaseID B15	Globally Unique Identifier of 16-byte length representing the SafetyBaseID of the Remote Safety Device to connect to. Note that B0 indicates the least significant byte.
12	SafetyBaseID B14	
13	SafetyBaseID B13	
14	SafetyBaseID B12	
15	SafetyBaseID B11	
16	SafetyBaseID B10	
17	SafetyBaseID B9	
18	SafetyBaseID B8	
19	SafetyBaseID B7	
20	SafetyBaseID B6	
21	SafetyBaseID B5	
22	SafetyBaseID B4	
23	SafetyBaseID B3	
24	SafetyBaseID B2	
25	SafetyBaseID B1	
26	SafetyBaseID B0	
27	SafetyProviderDelay B4	

28	SafetyProviderDelay B2	Unsigned 32-bit integer representing the SafetyProviderDelay of the Remote Safety Device to connect to in microseconds. Note that B0 indicates the least significant byte.
29	SafetyProviderDelay B1	
30	SafetyProviderDelay B0	
31	SafetyProviderLevel	SafetyProviderLevel of the Remote Safety Device

Table 49: Remote Safety Device Data Structure

6.4.5 SET REMOTE SAFETY DEVICE

A safety device shall accept a Set Remote Safety Device message with the data shown in Table 49. Note that this command shall only be accepted if the Platform Safety Interface is not Enabled. If the message is valid the Safety Device shall set the IP address, UDP port, SafetyProviderID, SafetyBaseID, SafetyProviderDelay, and SafetyProviderLevel of the Remote Safety to connect to and respond with a single data byte set to 0x00 for success. If rejected, a byte containing the error code shall be returned.

6.4.6 SET SAFETY CONSUMER ENABLE

A Platform Safety Interface shall accept a Set Safety Consumer Enable command in order to set the Enable input to the Safety Consumer in the Platform Safety Interface to a given state (Table 50). A single byte indicating success or an error code shall be returned.

Data	Data Type	Description
1	Enable	0=False, 1=True

Table 50: Safety Consumer Enable

6.4.7 GET SAFETY DIAGNOSTIC MESSAGES

A safety device shall respond to a Get Safety Diagnostic message with the data shown in Table 51. The diagnostic messages shall be represented using printable seven-bit ASCII codes terminated by a null (zero) byte. If rejected, a single byte containing the error code shall be returned.

Data	Data Type	Description
1	Num Messages	Only 2 are recorded. Valid values are 0, 1, 2.
..	Newest Message	ASCII text presenting diagnostic information
..	Second Newest Message	ASCII text presenting diagnostic information

Table 51: Safety Diagnostic Messages Data Structure

6.5 PAYLOAD CHANNEL UDP PROTOCOL

The Payload Channel Interface UDP Protocol defines the interface implemented by each Picatinny CLIK Payload Channel. For this protocol the Payload Channel is defined as the Server and shall only be transmitted in response to a received message. The following sections define the Picatinny CLIK defined message set that the Payload Channels shall accept and the possible responses. This message set can be expanded as permitted in 6.1. Table 52 identifies the payload messages defined by CLIK to include their specific message identifiers and whether or not their implementation is required.

Msg Id	Description	Data Description	Required
0x0000	Get Payload Channel Status	6.5.1	X
0x0001	Get Payload Channel Power Config	6.5.2	X
0x0002	Set Payload Channel Power	6.5.3	X
0x0003	Set Payload Channel Current Limit	6.5.4	

Table 52: Payload Channel UDP Message Identifiers

6.5.1 GET PAYLOAD CHANNEL STATUS

A Payload Channel shall respond to a Get Payload Channel Status message with the data shown in Table 53. An unsupported value shall be represented by the upper limits of the field (i.e. 255, 32767, 65535).

Data	Data Type	Description
1	Payload Connected	1=Yes 0=No
2	Voltage Out MSB	Unsigned 16-bit integer representing voltage provided to the payload in millivolts.
3	Voltage Out LSB	
4	Current Draw MSB	Signed 16-bit two's compliment integer representing platform current draw in milliamps. (negative = payload sourcing power)
5	Current Draw LSB	
6	Status Bit Fields	Bitwise state data. See Table 54

Table 53: Payload Channel Response Status Data Structure

Bit #	Name	Description	
0	Overcurrent Tripped	1=Yes	0=No
1	Channel Power state	1=On	0=Off
2	Safety Enable state	1=Activated	0=Not activated
3	Safety Execute state	1=Activated	0=Not activated
4	Reserved		
5	Reserved		
6	Reserved		
7	Reserved		

Table 54: Payload Channel Status Bit Fields

6.5.2 GET PAYLOAD CHANNEL POWER CONFIG

A Payload Channel shall respond to a Get Payload Channel Power Config message with the data shown in Table 55.

Data	Data Type	Description
1	Current Limit	Maximum payload current in 10ths of Amps.
2	Power Options	See Table 56

Table 55: Payload Channel Power Config Response Data Structure

Bit #	Name	Description
0	Overcurrent Circuit Reset	1=Yes 0=No
1	Configurable Current Limit	1=Yes 0=No
2	Reserved	
3	Reserved	
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	

Table 56: Payload Channel Power Option Bit Fields

6.5.3 SET PAYLOAD CHANNEL POWER

The Payload Channel shall accept a Set Payload Channel Power message with the data shown in Table 57. If the message is valid the Payload Channel shall set the payload power bus to the commanded state. If the Overcurrent Circuit Reset option is supported an overcurrent state shall be reset by setting the bus state to on. If the command is rejected a byte containing the error code shall be returned.

Data	Data Type	Description
1	Payload Channel Bus state	1=On 0=Off

Table 57: Set Platform Power Receive Data Structure

6.5.4 SET PAYLOAD CHANNEL CURRENT LIMIT

The Payload Channel shall optionally accept a Set Payload Channel Current Limit message with the data shown in Table 58. If message is valid the Payload Channel shall set the payload power bus to the commanded current limit. The upper limit shall be lesser of the platform current limit or connector current limit. If the command is rejected a byte containing the error code shall be returned.

Data	Data Type	Description
1	Payload Current Limit	Payload current limit in 10ths of Amps.

Table 58: Set Payload Channel Current Limit Receive Data Structure

APPENDIX A LITE ADAPTATION

A.1. SCOPE

A.1.1. PURPOSE

In cognizance of applications in which the requirements defined for mounting, power, signal and communication in the Design Standard may be a constraint regarding size, weight, and power needs, a Lite Adaptation was developed to address those applications. This Lite Adaptation is modeled after the Small Universal Platform Interface (sUPI) developed under U.S. Army Special Operations Command (SOCOM).

This Adaptation maintains the ability for integration of modular payloads onto UAS platforms with utilization of dedicated safety critical signals required to employ lethality but is tailored for one-to-one applications over the networked solution detailed in the main sections of the Design Standard.

A.1.1.1. SYSTEM OVERVIEW

The CLIK Lite architecture is a two-component system that establishes a standardized physical and electrical connection between a host platform and a payload. This one-to-one interface consists of:

- A Platform Side Interface
- A Payload Side Interface

This standard defines the physical specifications for mounting and the electrical pathways for power and data signals

The differences between CLIK Lite and CLIK are summarized in Table 59.

	CLIK Lite	CLIK
Mounting	Custom dovetail mount	Picatinny Rail
Payload weight capacity	5 lbs	Variable (Platform dependent)
Electrical connection	Spring-Loaded Contacts Integral to Mounting Interface	801 Series Connector
Platform Power	Raw power from 4S-6S Lithium Polymer battery pack	Platform must derive 28V from native power
Payload Current	4A Continuous	10A Continuous
Data Comms	USB 2.0 (Up to High Speed)	RS232 & Ethernet (10/100)
Discrete Signals	2x Safety Signals	2x Safety Signals
Platform to Payload Relationship	Peripheral	Networked

Table 59: CLIK to CLIK Lite Comparison

The CLIK Lite Interfaces between the Platform and Payload are depicted in Figure 29.

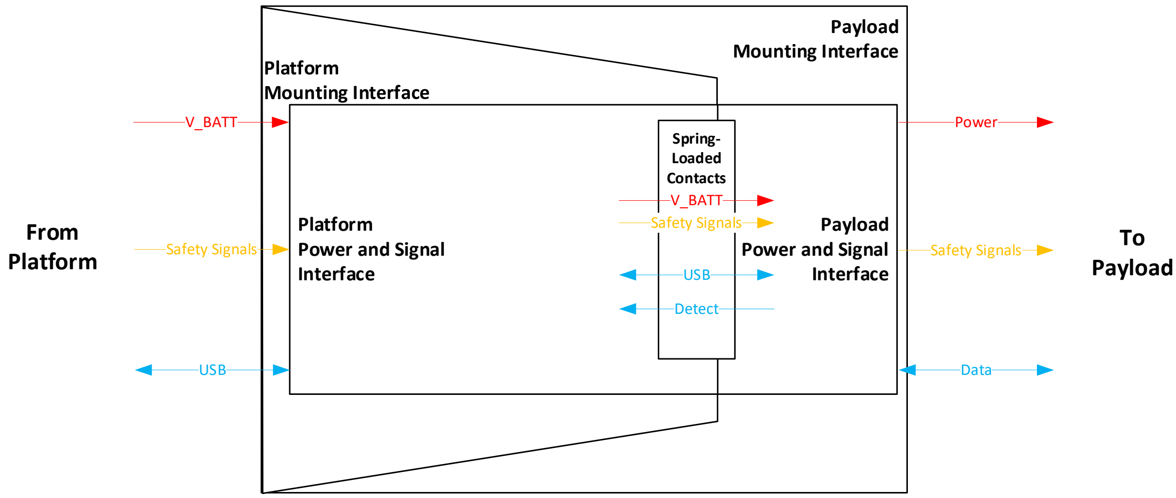


Figure 29: CLIK Lite Interface Overview

A.2. PAYLOAD REQUIREMENTS

A.2.1. MOUNTING/STRUCTURAL INTERFACE

The CLIK Lite Adaptation shall utilize the sUPI Payload Side Mechanical Interface for its physical mounting to the platform. For detailed implementation of this interface developers should request the latest sUPI Technical Data Package (TDP). Developers can use the TDP as provided to produce the module shown in Figure 30 or adapt the non-interface aspects of the design for their specific application.

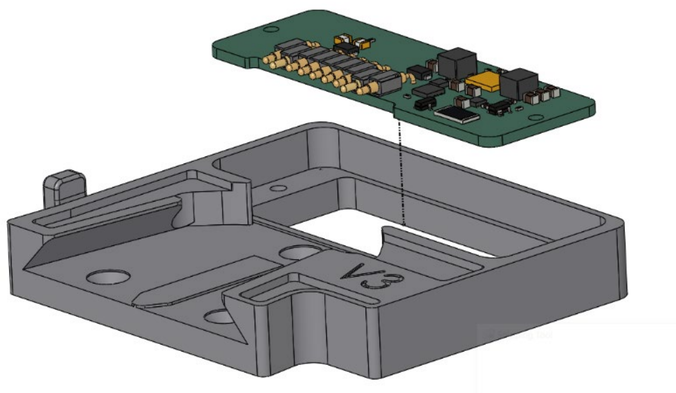


Figure 30: CLIK Lite Payload Side Interface

The Payload Side Interface is intended to be mounted to a payload via 3X M3x 0.5mm thread, 8mm long black-oxide stainless steel Phillips flat head screws (e.g. McMaster PN 91698A304). The mounting hole pattern is shown in Figure 31 below. Existing payloads should be modified in one of the following ways to install a Payload Side Interface:

- Drill and tap 4X holes into the payload as shown in Figure 31 in the desired location for mounting the Payload-Side Assembly. It is recommended that the payload CG align as closely as possible with the indicated point.
- Create an adapter that includes Payload Side Interface hole pattern and enables mounting to an existing mounting interface on a payload.

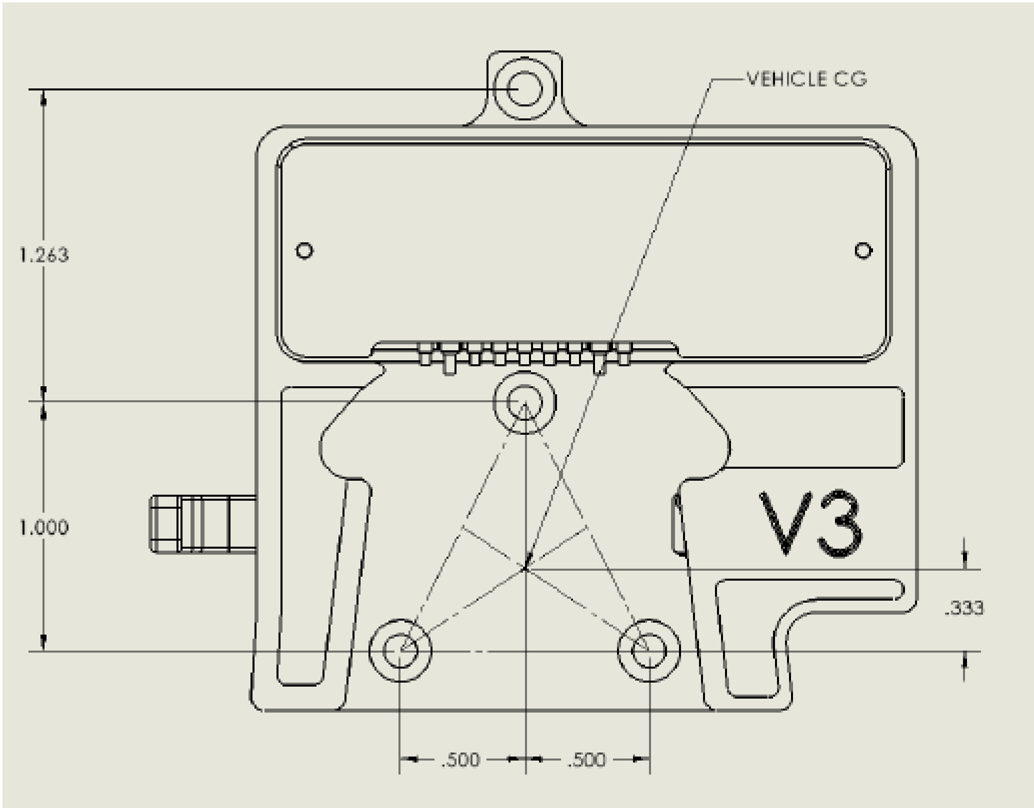


Figure 31: Payload Side Interface Hole Pattern

A.2.2. POWER AND SIGNAL INTERFACE

A.2.2.1. PHYSICAL CONNECTIONS

The Payload Side CCA shall utilize solder pads for connection wires from Payloads to the Payload Side Interface for the signals defined in Table 60 below along with corresponding Wire gauge range.

Signal	Wire Size (AWG)
VBATT	18 - 22
VBATT_RTN	18 - 22
V1	24 - 30
V1_RTN	24 - 30
V2	24 - 30
V2_RTN	24 - 30
SAFETY_ENABLE	24 - 30
SAFETY_ENABLE_RTN	24 - 30
SAFETY_EXECUTE	24 - 30
SAFETY_EXECUTE_RTN	24 - 30
DATA+	24 - 30
DATA-	24 - 30
GND	24 - 30

Table 60: Solder Pad Signals and Connection Sizes

The primary physical connection for Power and Signal between the Payload Side Interface and the Platform Side Interface shall be spring-loaded pogo pins with the following pin assignments shown in Table 61 below:

Pin	Signal
1	VBATT
2*	GND
3	DATA+
4	DATA-
5	SAFETY_ENABLE
6	SAFETY_EXECUTE
7	DETECT
8*	GND
9	VBATT

Table 61: Spring-loaded pogo Pin Signals and Assignments

**Pins 2 & 8 shall have an extended length to ensure GND signals make first contact before all other signals.*

A.2.2.2. POWER

The Payload Side Interface shall receive unregulated power from the Platform Side Interface, *VBATT*, which can range from 13VDC to 25.2VDC.

A.2.2.3. SAFETY SIGNALS

Similar to the Safety Signals defined in 3.2.8 in the Design Standard. Payloads shall utilize the Safety Enable and Safety Execute signals provided by the platform. A false value shall be interpreted as a signal with a voltage of less than or equal to 1.5V. A true value shall be interpreted as a voltage of greater than or equal to 3.5V.

It is the intention that both safety signals both be activated in order to permit the function to activate. The logical ANDing of the two independent signals ensures low probability of inadvertent activation. Note that the safety signals are a component of a payload's overall safety but are not exhaustive. Some payloads may require additional circuits, mechanisms, and/or commands to operate safely. The specific methods used to implement the full payload functionality are left to the developer. A notional example is depicted in Figure 32 where the enable signal is used to charge an energy storage circuit and the execute signal is used to release the energy into the ignition circuit.

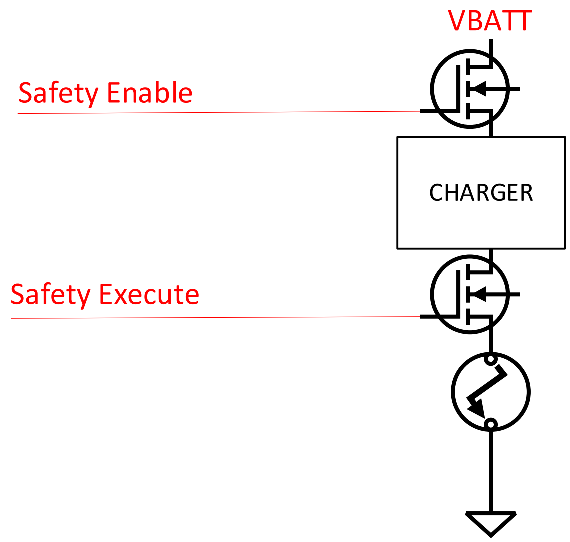


Figure 32: Notional CLIK Lite Payload Safety Circuit

A.2.2.4. DATA

A CLIK Lite Payload shall receive USB 2.0 3.3V Data+ and Data- Signals from the Platform. The Payload shall be configured as a peripheral *USB Device* with Class Code of *Communication Device Class (CDC)* selected. Under this configuration, the CLIK Lite Payload shall implement the protocol requirements defined in section *6.1 Common Message Protocol Format* and *6.2 Payload Serial Protocol*.

A.2.2.5. DETECT

The Payload Side Interface shall connect the *DETECT* signal to a 1-Wire compliant EEPROM. The contents of the EEPROM shall be provided as part of the TDP data.

A.3. PLATFORM REQUIREMENTS

A.3.1. MOUNTING/STRUCTURAL INTERFACE

The CLIK Lite Adaptation shall utilize the sUPI Platform Side Mechanical Interface for its physical mounting to the platform. For detailed implementation of this interface developers should request the latest sUPI Technical Data Package (TDP). Developers can use the TDP as provided to produce the module shown in Figure 33: CLIK Lite Platform Side Interface or adapt the non-interface aspects of the design for their specific application.

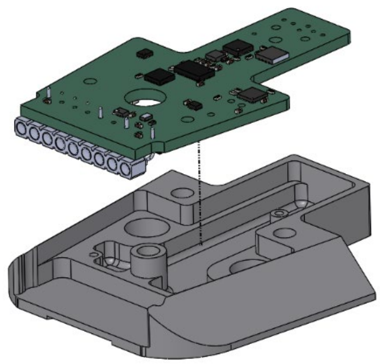


Figure 33: CLIK Lite Platform Side Interface

The Platform Side Interface is intended to be mounted to a platform via 3x M3x0.5 thread, 8mm long black-oxide stainless steel Phillips flat head screws (e.g. McMaster PN 91698A304). The mounting hole pattern is shown in Figure 34. Platforms should be modified in one of the following ways to install a Platform Side interface:

- Drill and tap 3x holes into the platform as shown in Figure 34 in the desired location for mounting the Platform Side
- Create an adapter that includes the platform side hole pattern and enables mounting to an existing mounting interface on a platform

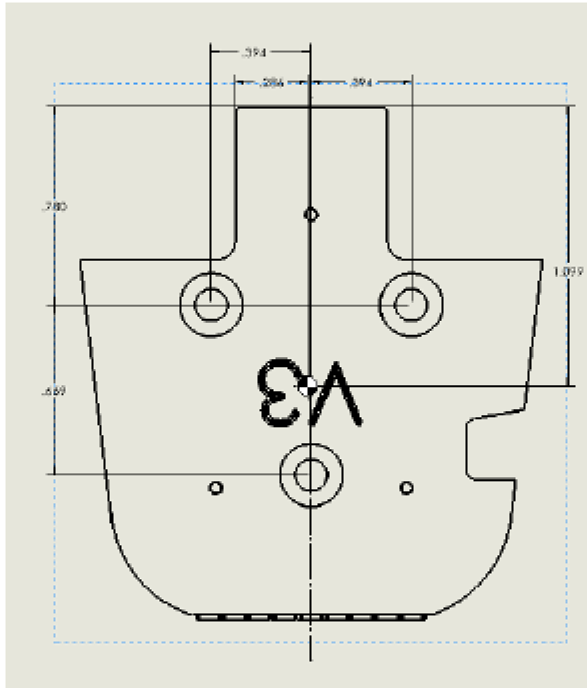


Figure 34: Platform Side Interface Hole Pattern

Important Note: Every effort should be made to maintain the platform's center of gravity (CG) in alignment with the CG reference point of the dovetail mechanism, thereby mirroring the payload's effort to align its CG with the dovetail mechanism reference point. While the standard mounting hole pattern is recommended to ensure interoperability, modifications to the hole pattern are permitted to meet platform-specific structural requirements. Such modifications should account for mission-specific stability, particularly when carrying payloads up to 5 lbs., and must ensure compatibility with the platform's structural analysis, CG alignment, and operational envelope.

A.3.2. POWER AND SIGNAL INTERFACE

A.3.2.1. PHYSICAL CONNECTIONS

The Platform Side CCA shall utilize solder pads for connection wires from the Platform to the Platform Side Interface for the signals defined in the Table 62 below along with corresponding Wire gauge range.

Signal	Wire Size (AWG)
VBATT	18 - 22
VBATT_RTN	18 - 22
SAFETY_ENABLE	24 - 30
SAFETY_EXECUTE	24 - 30
DATA+*	24 - 30
DATA-*	24 - 30
GND	24 - 30

Table 62: Solder Pad Signals and Connection Sizes

*Data+ and Data- Wires should be twisted with shield grounded on the platform recommended.

The primary physical connection for Power and Signal between the Platform Side Interface and the Payload Side Interface on the Platform Side Interface shall be contacts with the following assignments in the Table 63 below:

Pin	Signal
1	VBATT
2	GND
3	DATA+
4	DATA-
5	SAFETY_ENABLE
6	SAFETY_EXECUTE
7	DETECT
8	GND
9	VBATT

Table 63: Signal Contacts and Assignments

A.3.2.2. POWER

The Platform Side Interface shall receive native power from the platform, *VBATT*, which can range from 13VDC to 25.2VDC. *VBATT* shall be accessible to Payload Side Interface via contacts when mated.

The Platform Side CCA shall support the power options shown in Error! Reference source not found.:

Option	Behavior
*DETECT PAYLOAD**	Platform Side Interface communicates with Payload Side Interface via the *DETECT* Signal before *VBATT* Output turns ON. *VBATT* Output turns off when *DETECT* Signal is removed. If not set then VBATT is on at all times.
LO BATT CUTOFF	Platform Side Interface disables *VBATT* Output if Battery Voltage drops below a preconfigured amount.

Table 64: VBATT Output Options

**DETECT PAYLOAD* enabled is the recommended configuration

A.3.2.3. SAFETY SIGNALS

Similar to the Safety Signals defined in 4.2.2.3 in the main Design Standard, Platforms shall generate and distribute the Safety Enable and Safety Execute signals. Logic voltage output shall be 5V nominal with greater than or equal to 4.9V output for an active signal and less than or equal to .1V for an inactive signal. In a future revision of this document, a Serial Interface shall be added to allow cross-compatibility with the main Design Standard in regard to communication and function of the Safety Signals.

A.3.2.4. DATA

CLIK Lite shall provide USB 2.0 3.3V Data+ and Data- Signals from the Platform to the Payload. The Platform shall be configured as a USB Host. The Platform shall detect the Payload as a peripheral USB Device with *Communication Device Class (CDC)* Class Code and communicate with the Payload as defined in section 6.1 Common Message Protocol Format and 6.3 Payload Serial Protocol.

A.3.2.5. DETECT

The Platform Side Interface shall implement a detection circuit for detection of the Payload Side Interface when mated. This function is implemented by way of the *DETECT* Signal which reads and verifies the payload's EEPROM contents via 1-Wire interface.

Utilization of the *Detect* function may be used for controlling the *VBATT* Output as detailed in A.3.2.2.

APPENDIX B ACRONYMS AND ABBREVIATIONS

A	Amperes
AC	Armament Center
ACK	Acknowledgement
AiTR	Artificial Intelligence Target Recognition
ASCII	American Standard Code for Information Interchange
AWG	American Wire Gauge
BI_DA	Bi-directional Pair A
BI_DB	Bi-directional Pair B
C	Celsius
CAN	Canbus
CANH	Controller Area Network High
CANL	Controller Area Network Low
CCA	Circuit Card Assembly
CG	Center of Gravity
CIDR	Classless Inter-Domain Routing
CLIK	Common Lethality Integration Kit
cm	Centimeter
CNI	Channel N UDP Interface
CNS	Channel N Serial Interface
CRC	Cyclic Redundancy Check
CTI	Controlled Technical Information
CUI	Controlled Unclassified Information
CU	CLIK Unit
DC	Direct Current
DEVCOM	Combat Capabilities Development Command
DEVCOM-AC	Combat Capabilities Development Command Armament Center
EW	Electronic Warfare
FC	Fire Control
FCS	Fire Control Station
FSV	Fail-safe Substitute Values
GCS	Ground Control Station
GND	Ground
GPIO	General-Purpose Input/Output
GPS	Global Positioning System
GUID	Globally Unique Identifier
I/O	Input/Output (I/O)
I2C	Inter-Integrated Circuit Protocol
I^2C	Inter-Integrated Circuit Protocol
IAW	In Accordance With
ICD	Interface Control Document
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IO	Input Output

IP	Internet Protocol
ISR	Intelligence, Surveillance & Reconnaissance
JOTP	Joint Ordnance Test Procedure
LSB	Least Significant Byte
m	Meter
mA	milliamps
MAC	Maximum Aircraft Current
MCC	Maximum Connector Current
MIL	Military
MIL-DTL	Military Detail
MIL-STD	Military Standard
MIN	Minimum
mm	millimeter
MNR	MonitoringNumber
MPC	Maximum Payload Current
MRI	Machine-Readable Information
MSB	Most Significant Byte
ms	milliseconds
OC	On Center
OPC	Open Platform Communication
PBAS	Purpose Built Attritable System
PC	Payload Channels
PCB	Printed Circuit Board
PFH	Probability of Dangerous Failures per Hour
PI	Platform Interface
PKG	Package
PM	Project Manager
PM CCS	Project Manager Close Combat Systems
PM UAS	Project Manager Unmanned Aerial Systems
PN	Payload N Interface
QD	Quick Disconnect
REV	Revision
RF	Radio Frequency
s	seconds
SAPI	Safety Application Program Interface
SCL	Safety Communication Layer
SDA	Serial Data Line
SI	Safety Interface
SIL	Safety Integrity Level
SPDU	Safety Protocol Data Unit
SPI	Safety Parameter Interface
SRR	Short Range Reconnaissance
STD	Standard
sUPI	small Universal Payload Interface
TBD	To Be Determined
TDP	Technical Data Plan

TTL	Transistor-Transistor Logic
UA	Unified Architecture
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
UID	Unique Identifier
USB	Universal Serial Bus
UxS	Unmanned Systems
V	Volts
Ω	Ohms