# Pico Example

*Assembly Design Document*

## 1 Description

This example assembly is designed to run on the Raspberry Pi Pico. It includes a small collection of components that demonstrate how Adamant can be deployed onto a bare metal embedded system. In particular, the assembly demonstrates a simple rate group system, commanding, telemetry reporting using events, data products, and packets, and a simple fault detection and correction scheme. Explore the tables and diagrams below to learn more about the design of the example assembly.

## 2 Design

### 2.1 At a Glance

Below is a list of useful parameters and statistics that give a quick look into the makeup of the assembly.

- **Number of Components** - 26
- **Number of Component Types** - 23
- **Number of Active Components** - 7
- **Number of Passive Components** - 19
- **Number of Components with Queue** - 10
- **Number of Components without Queue** - 16
- **Number of Components with Events** - 21
- **Number of Components with Data Products** - 17
- **Number of Components with Data Dependencies** - *None*
- **Number of Components with Packets** - 11
- **Number of Components with Commands** - 17
- **Number of Components with Parameters** - 2
- **Number of Components with Faults** - 2
- **Number of Connections** - 134
- **Number of Events** - 148
- **Number of Data Products** - 38
- **Number of Data Dependencies** - *None*
- **Number of Packets** - 11
- **Number of Commands** - 54
- **Number of Parameters** - 6
- **Number of Faults** - 4

## 2.2 Components

All of the components in the Pico Example assembly are reusable components that can be found within the Adamant repository (in *src/components/*) with a few exceptions:

- **`Adc_Data_Collector`** - This component collects internal telemetry from the Raspberry Pi Pico including system voltage and temperature.
- **`Counter`** - This is an extremely simple component that produces an incrementing counter as telemetry.
- **`Fault_Producer`** - This is an component that can be used to inject a fault into the system via command.
- **`Oscillator`** - This is a simple component that produces two sinusoidal outputs into telemetry.

These specific component are located in the Example repository in *src/components/*. A full list of the components included in the assembly can be seen below. Note that components marked as active execute on their own task within the Ada runtime.

Table 1: Pico Example Components

| **Name** | **Type** | **Has Queue** | **Execution** |
|---|---|---|---|
| `System_Time_Instance` | `Gps_Time` | `no` | `passive` |
| `Ticker_Instance` | `Ticker` | `no` | `active` |
| `Tick_Divider_Instance` | `Tick_Divider` | `no` | `passive` |
| `Slow_Rate_Group` | `Rate_Group` | `yes` | `active` |
| `Fast_Rate_Group` | `Rate_Group` | `yes` | `active` |
| `Watchdog_Rate_Group` | `Rate_Group` | `yes` | `active` |
| `Ccsds_Command_ Depacketizer_Instance` | `Ccsds_Command_ Depacketizer` | `no` | `passive` |
| `Command_Router_Instance` | `Command_Router` | `yes` | `active` |
| `Event_Filter_Instance` | `Event_Filter` | `no` | `passive` |
| `Event_Limiter_Instance` | `Event_Limiter` | `no` | `passive` |
| `Event_Packetizer_ Instance` | `Event_Packetizer` | `no` | `passive` |
| `Product_Database_ Instance` | `Product_Database` | `no` | `passive` |
| `Product_Packetizer_ Instance` | `Product_Packetizer` | `yes` | `passive` |

| | | | |
|---|---|---|---|
| Ccsds_Packetizer_ Instance | Ccsds_Packetizer | no | passive |
| Ccsds_Serial_Interface_ Instance | Ccsds_Serial_Interface | yes | active |
| Counter_Instance | Counter | yes | passive |
| Oscillator_A | Oscillator | yes | passive |
| Oscillator_B | Oscillator | yes | passive |
| Adc_Data_Collector_ Instance | Adc_Data_Collector | no | passive |
| Zero_Divider_Instance | Zero_Divider | no | passive |
| Task_Watchdog_Instance | Task_Watchdog | no | passive |
| Fault_Producer_Instance | Fault_Producer | no | passive |
| Fault_Correction_ Instance | Fault_Correction | yes | active |
| Cpu_Monitor_Instance | Cpu_Monitor | no | passive |
| Queue_Monitor_Instance | Queue_Monitor | no | passive |
| Stack_Monitor_Instance | Stack_Monitor | no | passive |

Table 2: Pico Example Component Item Counts

| Component Name | Connectors | Commands | Events | Data Products | Data Dependencies | Parameters | Packets | Faults |
|---|---|---|---|---|---|---|---|---|
| System_Time_ Instance | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Ticker_Instance | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Tick_Divider_ Instance | 4 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| Slow_Rate_Group | 6 | 0 | 5 | 1 | 0 | 0 | 0 | 0 |
| Fast_Rate_Group | 6 | 0 | 5 | 1 | 0 | 0 | 0 | 0 |
| Watchdog_Rate_ Group | 6 | 0 | 5 | 1 | 0 | 0 | 0 | 0 |
| Ccsds_Command_ Depacketizer_ Instance | 8 | 1 | 7 | 2 | 0 | 0 | 1 | 0 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Command_Router_ Instance | 10 | 4 | 18 | 7 | 0 | 0 | 0 | 0 |
| Event_Filter_ Instance | 9 | 7 | 12 | 3 | 0 | 0 | 1 | 0 |
| Event_Limiter_ Instance | 9 | 8 | 14 | 3 | 0 | 0 | 1 | 0 |
| Event_ Packetizer_ Instance | 7 | 1 | 0 | 2 | 0 | 0 | 1 | 0 |
| Product_ Database_ Instance | 8 | 5 | 18 | 2 | 0 | 0 | 1 | 0 |
| Product_ Packetizer_ Instance | 7 | 4 | 9 | 0 | 0 | 0 | 1 | 0 |
| Ccsds_ Packetizer_ Instance | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Ccsds_Serial_ Interface_ Instance | 4 | 0 | 3 | 0 | 0 | 0 | 0 | 0 |
| Counter_Instance | 6 | 3 | 6 | 0 | 0 | 0 | 1 | 0 |
| Oscillator_A | 7 | 3 | 6 | 1 | 0 | 3 | 0 | 0 |
| Oscillator_B | 7 | 3 | 6 | 1 | 0 | 3 | 0 | 0 |
| Adc_Data_ Collector_ Instance | 3 | 0 | 0 | 3 | 0 | 0 | 0 | 0 |
| Zero_Divider_ Instance | 4 | 1 | 3 | 0 | 0 | 0 | 1 | 0 |
| Task_Watchdog_ Instance | 9 | 4 | 10 | 4 | 0 | 0 | 0 | 2 |
| Fault_Producer_ Instance | 5 | 2 | 3 | 0 | 0 | 0 | 0 | 2 |
| Fault_ Correction_ Instance | 7 | 5 | 11 | 4 | 0 | 0 | 0 | 0 |
| Cpu_Monitor_ Instance | 7 | 1 | 2 | 1 | 0 | 0 | 1 | 0 |
| Queue_Monitor_ Instance | 7 | 1 | 2 | 1 | 0 | 0 | 1 | 0 |
| Stack_Monitor_ Instance | 7 | 1 | 2 | 1 | 0 | 0 | 1 | 0 |

Component Descriptions:

- **System_Time_Instance** - This component provides the system time for the assembly.
- **Ticker_Instance** - This component uses a periodic signal to drive the assembly rate groups.
- **Tick_Divider_Instance** - This component divides a periodic signal into intervals suitable for the assembly rate groups.
- **Slow_Rate_Group** - This component provides a 0.5 Hz task for other components to execute on periodically.
- **Fast_Rate_Group** - This component provides a 5 Hz task for other components to execute on periodically.

- **Watchdog_Rate_Group** - This component provides a 1 Hz task for the watchdog components to execute on periodically.
- **Ccsds_Command_Depacketizer_Instance** - This component converts CCSDS packets containing commands to valid Adamant formatted command types.
- **Command_Router_Instance** - This component provides command and command response routing throughout the assembly.
- **Event_Filter_Instance** - This component filters events by ID.
- **Event_Limiter_Instance** - This component filters out events that spam the system too frequently.
- **Event_Packetizer_Instance** - This component gathers events and packetizes them for downlink.
- **Product_Database_Instance** - This component serves as the database for data products throughout the system.
- **Product_Packetizer_Instance** - This component periodically fetches values from the Product Database and packetizes them for downlink.
- **Ccsds_Packetizer_Instance** - This component converts Adamant formatted packets to CCSDS for downlink.
- **Ccsds_Serial_Interface_Instance** - This component manages the Raspberry Pi Pico UART interface for command uplink and telemetry downlink.
- **Counter_Instance** - This component periodically produces a count data product.
- **Oscillator_A** - This component periodically produces an oscillating data product.
- **Oscillator_B** - This component periodically produces an oscillating data product.
- **Adc_Data_Collector_Instance** - This component collects data from the internal Raspberry Pi Pico analog to digital converted (ADC) and reports it as telemetry.
- **Zero_Divider_Instance** - This component responds to a command that divides by zero if received. This can be used to trigger a fault condition within the processor.
- **Task_Watchdog_Instance** - This component monitors other critical active components (tasks) within the assembly to make sure they continue to run.
- **Fault_Producer_Instance** - This component can be used to induce a fault into the system by command.
- **Fault_Correction_Instance** - This component produces a corrective action (a command) for any fault that is thrown in the system.
- **Cpu_Monitor_Instance** - This component produces a packet that includes the CPU usage percentage for each task and interrupt in the system.
- **Queue_Monitor_Instance** - This component produces a packet that includes the current and maximum queue usage for each component in the system.
- **Stack_Monitor_Instance** - This component produces a packet that includes the maximum stack and secondary stack usage for each component in the system.

## 2.3 Views

This section shows the example assembly visually as a set of *views*. Each shows a specific set of components and connections (while not showing other components and connections) in order to highlight a particular function of the assembly. Components that are bold are *active*, meaning they have an Ada task assigned to them. Connections are labeled with the type that is passed along them. A dotted line indicates that the connection is asynchronous, meaning the data is put onto a queue for later processing. A solid line indicates that the connection is synchronous, meaning processing of that data occurs right when the data is passed along the connector.

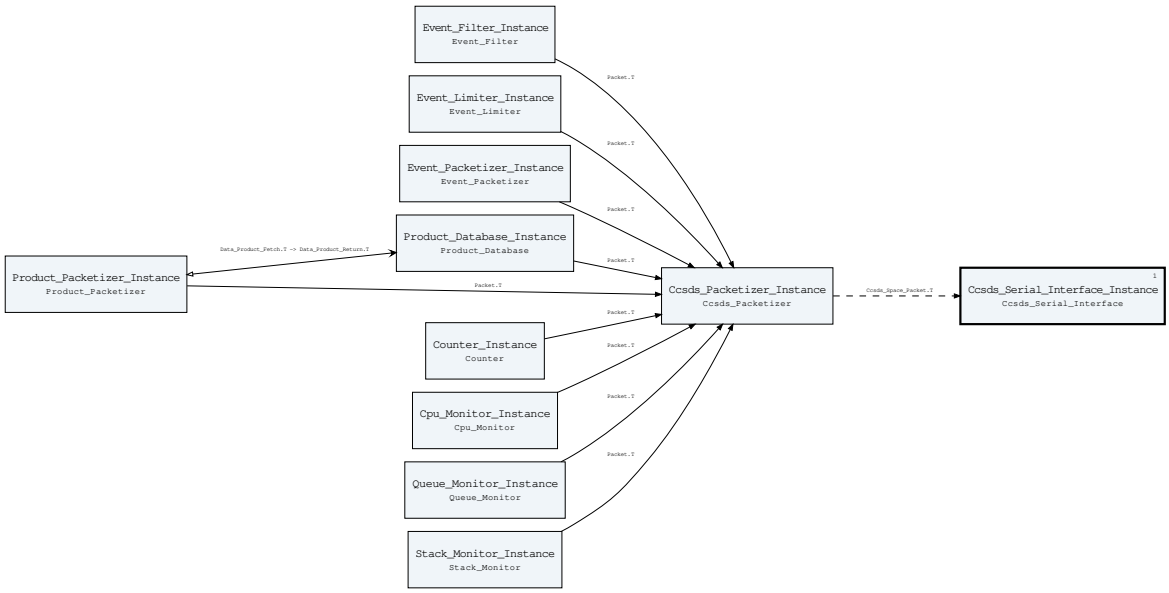


Figure 1: **Downlink View:** Any packet produced by the assembly is sent to the Ccsds_Packetizer_Instance. This component takes the Adamant formatted packets, converts them to CCSDS, and sends them asynchronously to the Ccsds_Serial_Interface_Instance. The serial interface component will take packets stored on its queue and transmit them over the Raspberry Pi Pico UART. The Product_Packetizer_Instance periodically requests data products from the Product_Database_Instance to create packets containing data from multiple components. These packets are also forwarded to the Ccsds_Packetizer_Instance for downlink.


Figure 2: **Uplink View:** The Ccsds_Serial_Interface_Instance receives data on the Raspberry Pi Pico UART. It is actively looking for a sync pattern followed by a CCSDS packet header. If it receives a valid CCSDS packet it forwards it along to the Ccsds_Commmand_Depacketizer_Instance. This component looks for commands in the CCSDS packet, extracts them, and then forwards them along the Command_Router_Instance for routing and later execution.

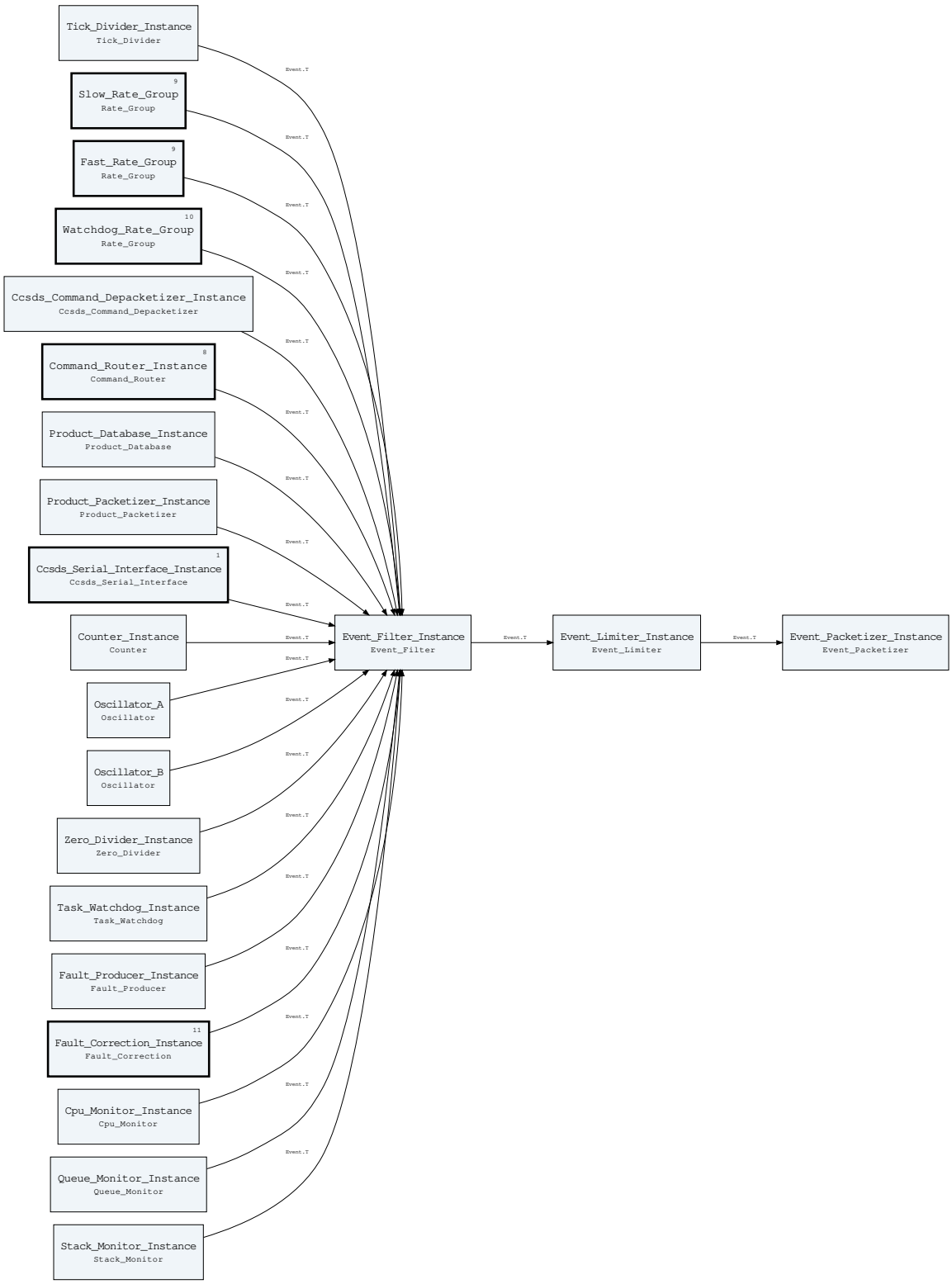


Figure 3: **Events View:** Events are produced by components when something interesting happens. All components send their events first to the Event_Filter_Instance. This component can enabled/disable individual events by ID. Next, events are forwarded to the Event_Limiter_Instance. This component will start limiting problematic spam events that might flood the system. Any event that passes both of these filtering components is passed to the Event_Packetizer_Instance for collection in a packet for later downlink.

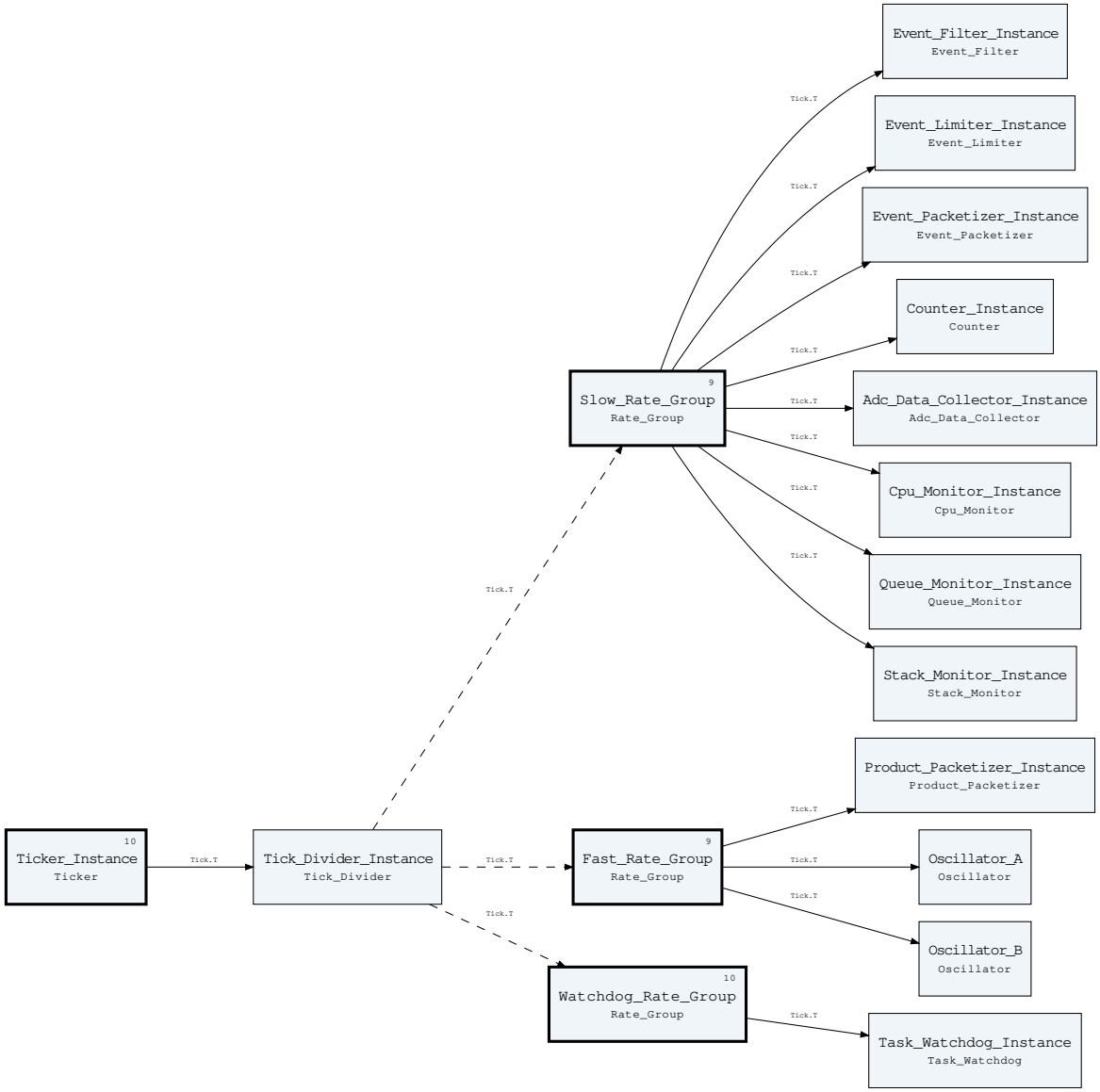


Figure 4: **Tick View:** In this assembly, Tick.T types are used to indicate when periodic tasks should be run. This view shows the rate groups that exist in the assembly. The Ticker_Instance runs in a high priority task and produces a Tick.T at 5 Hz. This tick is forwarded to the Tick_Divider_Instance which takes the 5 Hz signal and divides it up among 3 different rate groups. The Slow_Rate_Group runs at 0.5 Hz. The Fast_Rate_Group runs a 5 Hz. The Watchdog_Rate_Group runs as 1 Hz. Each of these rate group components will call the components connected to them every time they receive a tick from upstream. This architecture allows for periodic execution of certain functions, such as collecting telemetry or creating packets. Note that the actual execution of each rate group is also interleaved by the Ada scheduler. Each rate group is an active component, which executes on a task with a priority shown in the top right hand corner of the component. Rate groups produce telemetry relating how long it takes them to execute, both in CPU time and in wall clock time.

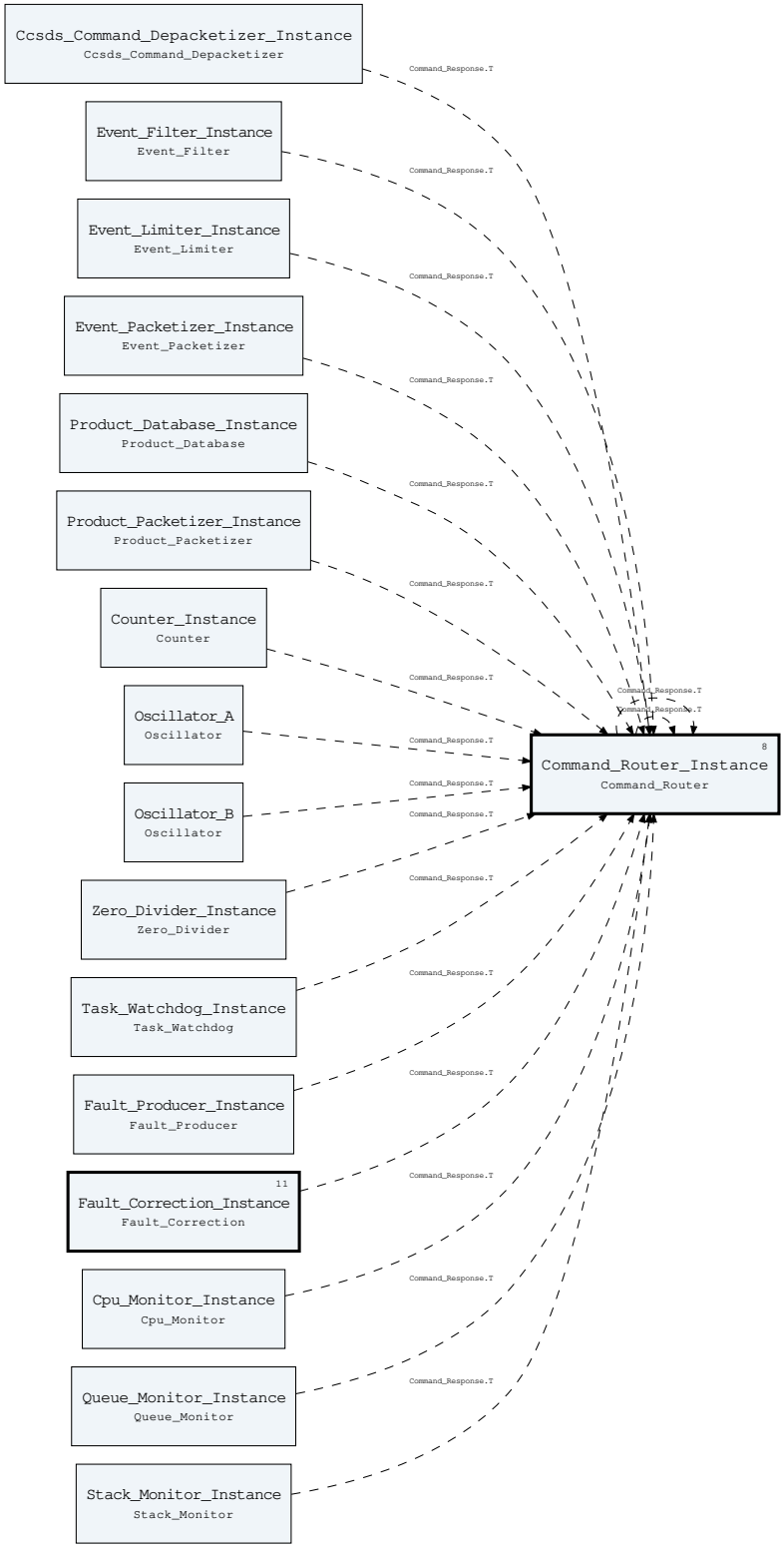


Figure 5: **Command Response View:** After a component executes a command received from the Command_Router_Instance, it passes back a Command_Response.T type to the router letting it know if the command succeeded or failed. This connection is also used to register all the commands with the router an initialization. This allows the Command_Router_Instance to create the internal routing table that it uses to route incoming commands.

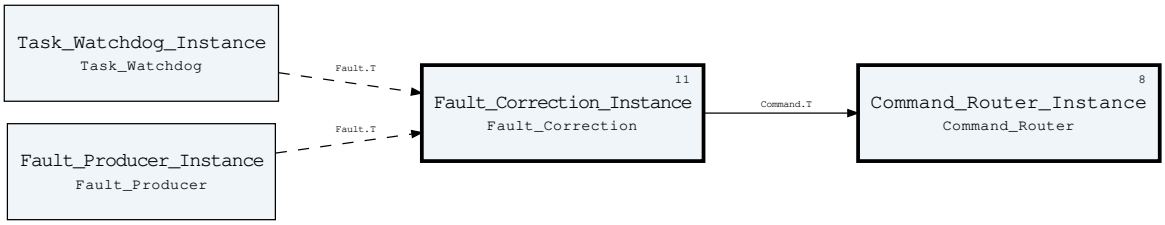


Figure 6: **Faults View:** Any component that produces faults will send them to the Fault_Correction_Instance. This component maps the fault to a response action. This action is in the form of a correction command which is sent to the Command_Router_Instance synchronously for execution. In the Raspberry Pi Pico assembly, two components can throw faults. The first is the Task_Watchdog_Instance. This component monitors some critical tasks to ensure that they are always running in a timely fashion. If one stops executing, a fault is thrown. The Fault_Producer_Instance is a simple component that throws a fault when commanded to. This can be used to inject a fault into the system for testing purposes.

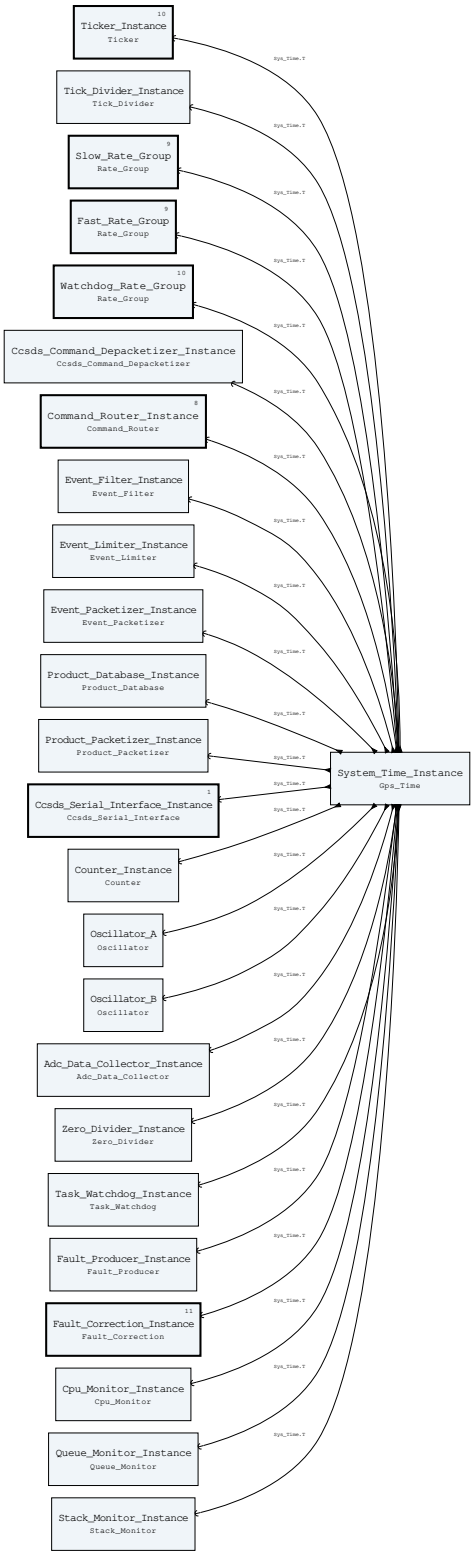


Figure 7: **Time View:** All components in the system fetch time by requesting it from the System_Time_Instance. By implementing the time system at the Adamant architectural level, different time sources and time synchronization schemes can be easily swapped into the system by replacing System_Time_Instance with a more tailored version. Note that synchronous connectors exhibit extremely low overhead, so exposing anything at the architectural level should not be prohibitive in terms of performance.

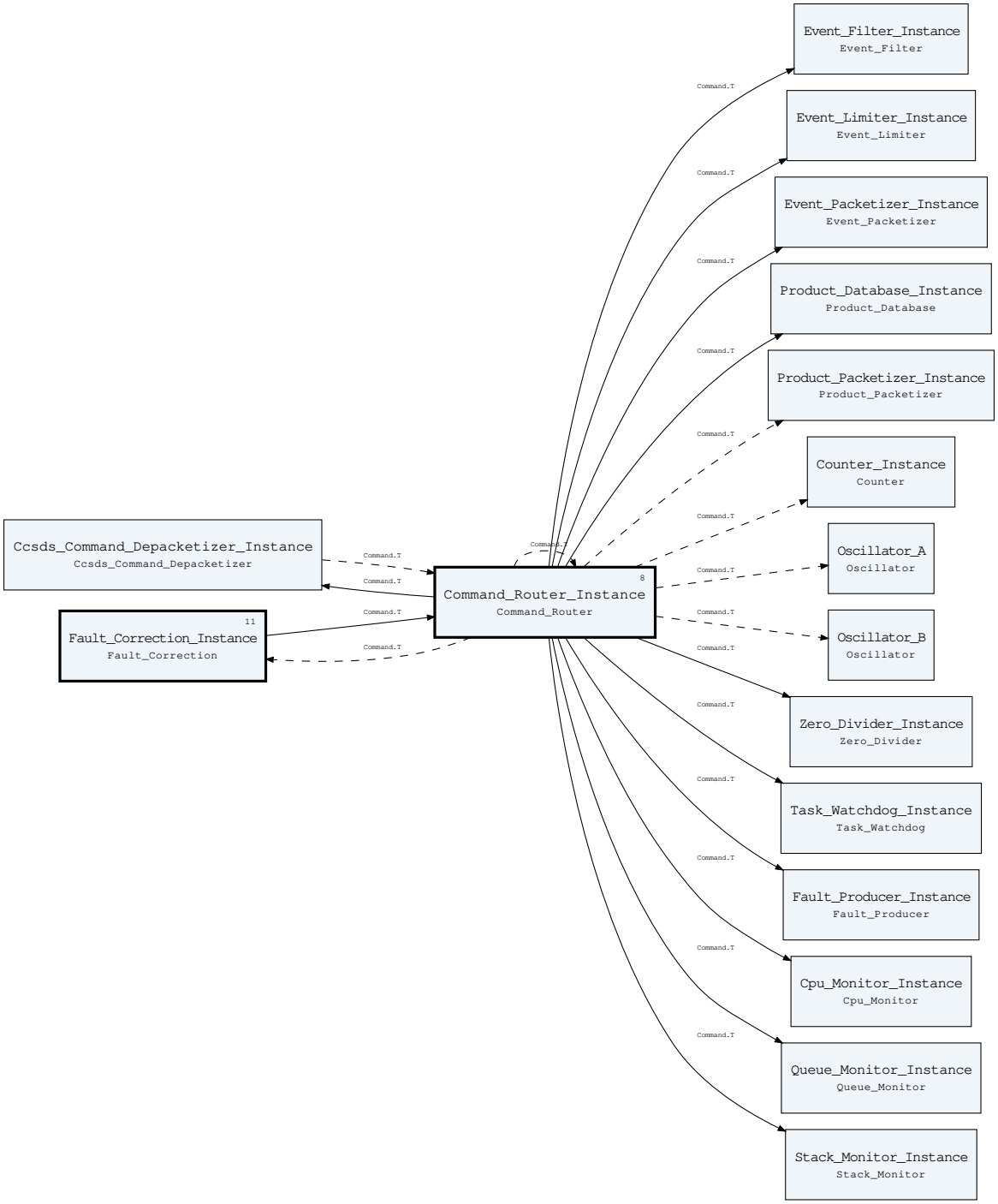


Figure 8: **Command View:** This view shows how commands are routed throughout the assembly. Commands are contained in the Command.T data type. Commands coming from the ground system originate from the Ccsds_Command_Depacketizer_Instance and then get passed the Command_Router_Instance. The router looks at the command's ID, determines the destination component for which the command is intended, and then forwards the command to that appropriate destination component. When a destination component receives a command, it will execute it and pass a Command_Response.T data type back to the command router (shown in the Command Response View). Note that the Fault_Correction_Instance can also produce commands in order to correct a system fault. Commands from the Fault_Correction_Instance are passed to the router synchronously, bypassing the standard command queue that the Ccsds_Command_Depacketizer_Instance uses.

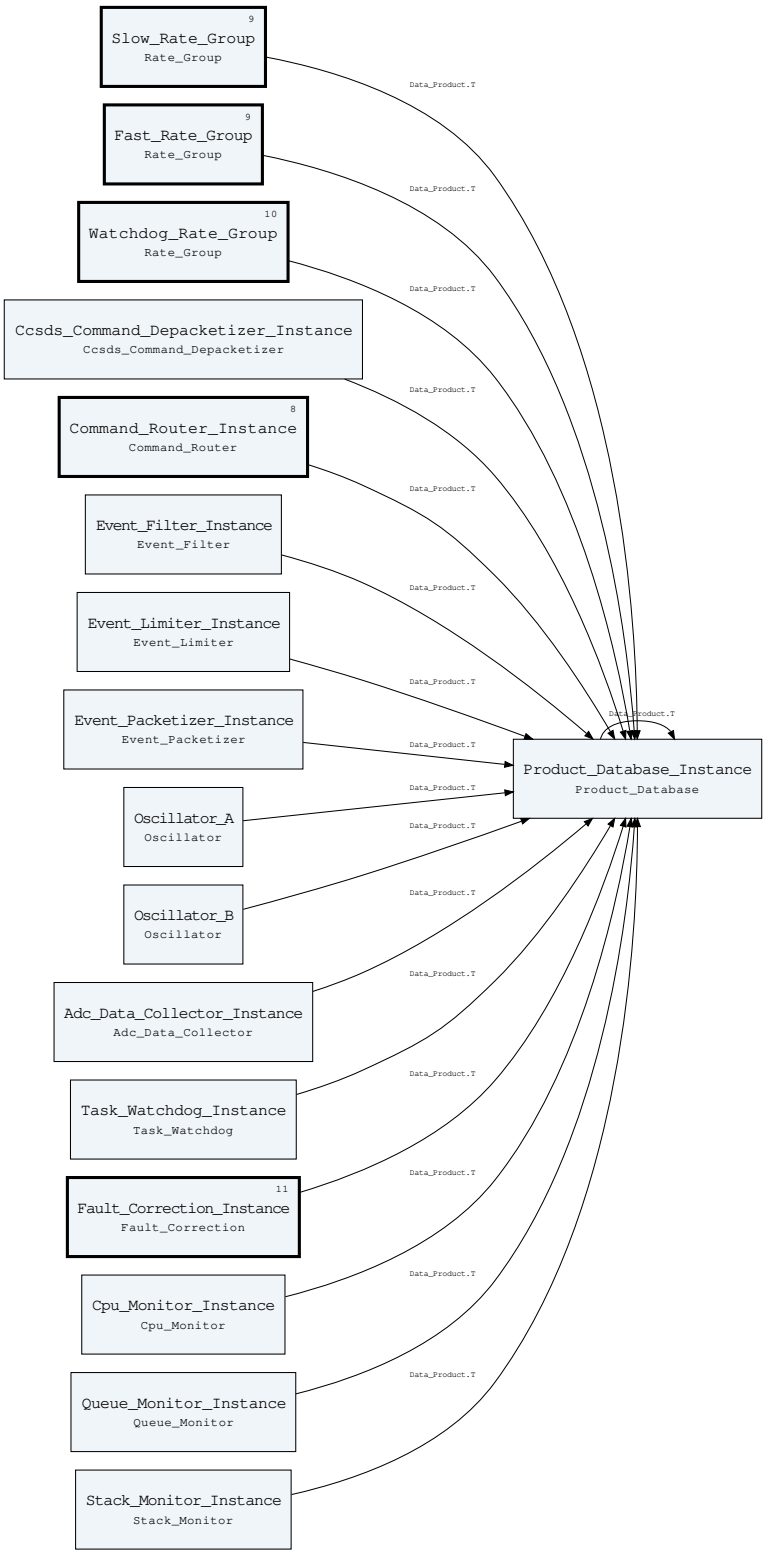


Figure 9: **Data Products View:** Any data product produced by a component during execution is sent synchronously to on onboard storage database, the Product_Database_Instance. This component stores the latest value and timestamp of each data product in the system. Other components, such as the Product_Packetizer_Instance can then fetch these data products at a later time for limit checking, packetization, etc.

## 2.4 Task Priorities

The table below outlines the system tasks for the Pico Example assembly. Task names are in the form *component_ name.task_ name*. The priority *rank* is a number from 1 to $n$ denoting how the priority of a component's task compares to others in the system. A rank of 1 is the highest priority in the system. The *priority value* is the actual priority number provided to the system scheduler. A larger *priority value* value signifies a higher priority task.

Table 3: Pico Example Component Task Priorities

| **Task Number** | **Task Name** | **Priority Rank** | **Priority Value** |
|---|---|---|---|
| 0 | Fault_Correction_Instance.Active_Task | 1 | 11 |
| 1 | Ticker_Instance.Active_Task | 2 | 10 |
| 2 | Watchdog_Rate_Group.Active_Task | 2 | 10 |
| 3 | Slow_Rate_Group.Active_Task | 3 | 9 |
| 4 | Fast_Rate_Group.Active_Task | 3 | 9 |
| 5 | Command_Router_Instance.Active_Task | 4 | 8 |
| 6 | Ccsds_Serial_Interface_Instance.Active_Task | 5 | 1 |
| 7 | Ccsds_Serial_Interface_Instance.Listener_ Task | 6 | 0 |

## 2.5 Commands

The table below shows the commands for the Pico Example assembly.

Table 4: Pico Example Commands

| **Command ID** | **Command Name** | **Argument Type** |
|---|---|---|
| 0x0001 (1) | Ccsds_Command_Depacketizer_ Instance.Reset_Counts | - |
| 0x0002 (2) | Command_Router_Instance. Noop | - |
| 0x0003 (3) | Command_Router_Instance. Noop_Arg | Command_Router_Arg.T |
| 0x0004 (4) | Command_Router_Instance. Noop_Response | - |

| | | |
|---|---|---|
| 0x0005 (5) | Command_Router_Instance.<br>Reset_Data_Products | - |
| 0x0006 (6) | Event_Filter_Instance.<br>Filter_Event | Event_Filter_Single_Event_<br>Cmd_Type.T |
| 0x0007 (7) | Event_Filter_Instance.<br>Unfilter_Event | Event_Filter_Single_Event_<br>Cmd_Type.T |
| 0x0008 (8) | Event_Filter_Instance.<br>Filter_Event_Range | Event_Filter_Id_Range.T |
| 0x0009 (9) | Event_Filter_Instance.<br>Unfilter_Event_Range | Event_Filter_Id_Range.T |
| 0x000a (10) | Event_Filter_Instance.<br>Enable_Event_Filtering | - |
| 0x000b (11) | Event_Filter_Instance.<br>Disable_Event_Filtering | - |
| 0x000c (12) | Event_Filter_Instance.Dump_<br>Event_States | - |
| 0x000d (13) | Event_Limiter_Instance.<br>Enable_Event_Limit | Event_Single_State_Cmd_<br>Type.T |
| 0x000e (14) | Event_Limiter_Instance.<br>Disable_Event_Limit | Event_Single_State_Cmd_<br>Type.T |
| 0x000f (15) | Event_Limiter_Instance.<br>Enable_Event_Limit_Range | Event_Limiter_Id_Range.T |
| 0x0010 (16) | Event_Limiter_Instance.<br>Disable_Event_Limit_Range | Event_Limiter_Id_Range.T |
| 0x0011 (17) | Event_Limiter_Instance.<br>Enable_Event_Limiting | - |
| 0x0012 (18) | Event_Limiter_Instance.<br>Disable_Event_Limiting | - |
| 0x0013 (19) | Event_Limiter_Instance.Set_<br>Event_Limit_Persistence | Event_Limiter_Persistence_<br>Type.T |
| 0x0014 (20) | Event_Limiter_Instance.<br>Dump_Event_States | - |
| 0x0015 (21) | Event_Packetizer_Instance.<br>Send_Packet | - |

| | | |
|---|---|---|
| 0x0016 (22) | Product_Database_Instance.<br>Clear_Override | Data_Product_Id.T |
| 0x0017 (23) | Product_Database_Instance.<br>Clear_Override_For_All | - |
| 0x0018 (24) | Product_Database_Instance.<br>Override | Data_Product.T |
| 0x0019 (25) | Product_Database_Instance.<br>Dump | Data_Product_Id.T |
| 0x001a (26) | Product_Database_Instance.<br>Dump_Poly_Type | Data_Product_Poly_Extract.T |
| 0x001b (27) | Product_Packetizer_<br>Instance.Set_Packet_Period | Packet_Period.T |
| 0x001c (28) | Product_Packetizer_<br>Instance.Enable_Packet | Packet_Id.T |
| 0x001d (29) | Product_Packetizer_<br>Instance.Disable_Packet | Packet_Id.T |
| 0x001e (30) | Product_Packetizer_<br>Instance.Send_Packet | Packet_Id.T |
| 0x001f (31) | Oscillator_A.Set_Frequency | Packed_F32.T |
| 0x0020 (32) | Oscillator_A.Set_Amplitude | Packed_F32.T |
| 0x0021 (33) | Oscillator_A.Set_Offset | Packed_F32.T |
| 0x0022 (34) | Oscillator_B.Set_Frequency | Packed_F32.T |
| 0x0023 (35) | Oscillator_B.Set_Amplitude | Packed_F32.T |
| 0x0024 (36) | Oscillator_B.Set_Offset | Packed_F32.T |
| 0x0025 (37) | Zero_Divider_Instance.<br>Divide_By_Zero | Packed_U32.T |
| 0x0026 (38) | Task_Watchdog_Instance.<br>Enable_Watchdog_Pet_Checks | - |
| 0x0027 (39) | Task_Watchdog_Instance.<br>Disable_Watchdog_Pet_Checks | - |
| 0x0028 (40) | Task_Watchdog_Instance.Set_<br>Watchdog_Limit | Pico_Example_Task_Watchdog_<br>List_Watchdog_Limit_Cmd.T |

| | | |
|---|---|---|
| 0x0029 (41) | Task_Watchdog_Instance.Set_Watchdog_Action | Pico_Example_Task_Watchdog_List_Watchdog_Action_Cmd.T |
| 0x002a (42) | Fault_Producer_Instance.Throw_Fault_1 | - |
| 0x002b (43) | Fault_Producer_Instance.Throw_Fault_2 | - |
| 0x002c (44) | Fault_Correction_Instance.Enable_Fault_Response | Pico_Example_Fault_Responses_Packed_Id_Type.T |
| 0x002d (45) | Fault_Correction_Instance.Disable_Fault_Response | Pico_Example_Fault_Responses_Packed_Id_Type.T |
| 0x002e (46) | Fault_Correction_Instance.Clear_Fault_Response | Pico_Example_Fault_Responses_Packed_Id_Type.T |
| 0x002f (47) | Fault_Correction_Instance.Clear_All_Fault_Responses | - |
| 0x0030 (48) | Fault_Correction_Instance.Reset_Data_Products | - |
| 0x0031 (49) | Cpu_Monitor_Instance.Set_Packet_Period | Packed_U16.T |
| 0x0032 (50) | Queue_Monitor_Instance.Set_Packet_Period | Packed_U16.T |
| 0x0033 (51) | Stack_Monitor_Instance.Set_Packet_Period | Packed_U16.T |
| 0x0063 (99) | Counter_Instance.Set_Count | Packed_U32.T |
| 0x0064 (100) | Counter_Instance.Reset_Count | - |
| 0x0065 (101) | Counter_Instance.Set_Count_Add | Operands.T |

Command Descriptions:

- **Ccsds_Command_Depacketizer_Instance.Reset_Counts** - This command resets the internal counts for the data products.
- **Command_Router_Instance.Noop** - Simple NOOP command which produces an event saying that it was triggered. This can be used to self test the command routing system and verify system aliveness.
- **Command_Router_Instance.Noop_Arg** - Simple NOOP command which produces an event saying that it was triggered with a certain Arg. This can be used to self test the command argument validation system. Sending a command with an Arg value of 868 will cause the component

to Fail the command. Any other value will produce a successfully executed command.

- **Command_Router_Instance.Noop_Response** - A NOOP command which self tests the command response forwarding mechanism. The command handler itself acts as a command sender component, and sends out a NOOP command with a registered Source Id. The Command Router should then send out an event saying that the command response was forwarded and received again by the Command Router.
- **Command_Router_Instance.Reset_Data_Products** - This command resets the values of all the component's data product to the values at initialization.
- **Event_Filter_Instance.Filter_Event** - Enable the event filtering for a specific event ID.
- **Event_Filter_Instance.Unfilter_Event** - Disable the event filtering for a specific event ID.
- **Event_Filter_Instance.Filter_Event_Range** - Enable the event filtering for a specific range of event IDs.
- **Event_Filter_Instance.Unfilter_Event_Range** - Disable the event filtering for a specific range of event IDs.
- **Event_Filter_Instance.Enable_Event_Filtering** - Enable the component to filter events that have been set to be filtered.
- **Event_Filter_Instance.Disable_Event_Filtering** - Disable the component so that all events will not be filtered. The event states will be maintained for when re-enabled.
- **Event_Filter_Instance.Dump_Event_States** - Dump a packet for the state of all events pertaining to if they are filtered or not.
- **Event_Limiter_Instance.Enable_Event_Limit** - Enable the event limiter for a specific event ID.
- **Event_Limiter_Instance.Disable_Event_Limit** - Disable the event limiter for a specific event ID.
- **Event_Limiter_Instance.Enable_Event_Limit_Range** - Enable the event limiter for a specific range of event IDs.
- **Event_Limiter_Instance.Disable_Event_Limit_Range** - Disable the event limiter for a specific range of event IDs.
- **Event_Limiter_Instance.Enable_Event_Limiting** - Enable the component to limit events that have been set to be limited.
- **Event_Limiter_Instance.Disable_Event_Limiting** - Disable the component so that all events will not be limited. The event states and counts will be maintained for when re-enabled.
- **Event_Limiter_Instance.Set_Event_Limit_Persistence** - Change the persistence of the event limiter for all events that are limited. Value must be between 1 and 7.
- **Event_Limiter_Instance.Dump_Event_States** - Dump a packet for the state of all events on if they are limited or not.
- **Event_Packetizer_Instance.Send_Packet** - Send a packet out next tick, unless there are no events stored within the component.
- **Product_Database_Instance.Clear_Override** - Clear the override condition for the data product of the provided ID.
- **Product_Database_Instance.Clear_Override_For_All** - Clear the override condition for all data products in the product store.
- **Product_Database_Instance.Override** - Override the value of a data product in the data product store. The value of this data product will be fixed to the commanded value, ignoring all other updates, until the override is cleared.

- **Product_Database_Instance.Dump** - Dump the data product of the provided ID in a packet.
- **Product_Database_Instance.Dump_Poly_Type** - Dump the data product of the provided ID into a poly type based on the provided offset and length.
- **Product_Packetizer_Instance.Set_Packet_Period** - Command to change the period of packet generation for a given packet id.
- **Product_Packetizer_Instance.Enable_Packet** - Command to enable the emission of a packet from the packetizer.
- **Product_Packetizer_Instance.Disable_Packet** - Command to disable the emission of a packet from the packetizer.
- **Product_Packetizer_Instance.Send_Packet** - Command to build specific packet and send it out on the next available tick. The packet is built and sent regardless of the packet being enabled or disabled.
- **Oscillator_A.Set_Frequency** - Set the frequency of the oscillator in Hz
- **Oscillator_A.Set_Amplitude** - Set the amplitude of the oscillator
- **Oscillator_A.Set_Offset** - Set the Y offset of the oscillator
- **Oscillator_B.Set_Frequency** - Set the frequency of the oscillator in Hz
- **Oscillator_B.Set_Amplitude** - Set the amplitude of the oscillator
- **Oscillator_B.Set_Offset** - Set the Y offset of the oscillator
- **Zero_Divider_Instance.Divide_By_Zero** - You must provide the correct magic number as argument to this command for it to be executed.
- **Task_Watchdog_Instance.Enable_Watchdog_Pet_Checks** - Command to enable the watchdog component to check all connected components for incoming pets.
- **Task_Watchdog_Instance.Disable_Watchdog_Pet_Checks** - Command to disable the watchdog component to check all connected components for incoming pets.
- **Task_Watchdog_Instance.Set_Watchdog_Limit** - Set the limit value for the watchdog given an index and the new index value.
- **Task_Watchdog_Instance.Set_Watchdog_Action** - Sets the action of a petter given the index of that petter and the updated action. Note that actions cannot be promoted to fault if they were not provided a fault id.
- **Fault_Producer_Instance.Throw_Fault_1** - Throw the first fault.
- **Fault_Producer_Instance.Throw_Fault_2** - Throw the second fault.
- **Fault_Correction_Instance.Enable_Fault_Response** - Enable a fault response for the provided ID. This will only succeed if another response with the same Fault ID is not already enabled.
- **Fault_Correction_Instance.Disable_Fault_Response** - Disable a fault response for the provided ID.
- **Fault_Correction_Instance.Clear_Fault_Response** - Resets a fault response to the Enabled state of the provided ID. If the fault is latched, it unlatches the fault.
- **Fault_Correction_Instance.Clear_All_Fault_Responses** - Resets all fault responses to the Enabled state. Unlatches all latched fault responses.
- **Fault_Correction_Instance.Reset_Data_Products** - This command resets the values of all the component's data product to the values at initialization, except for the Fault_Response_Statuses data product which can be reset by the Clear_All_Fault_Responses command.
- **Cpu_Monitor_Instance.Set_Packet_Period** - Set the period of the packet. A period of zero disables the sending of the packet.

- **Queue_Monitor_Instance.Set_Packet_Period** - Set the period of the packet. A period of zero disables the sending of the packet.
- **Stack_Monitor_Instance.Set_Packet_Period** - Set the period of the packet. A period of zero disables the sending of the packet.
- **Counter_Instance.Set_Count** - Change the current counter value in the counter component
- **Counter_Instance.Reset_Count** - Reset the current counter value in the counter component to zero
- **Counter_Instance.Set_Count_Add** - Change the current counter value in the counter component to the sum of the arguments

## 2.6 Parameters

The table below shows the parameters for the Pico Example assembly.

Table 5: Pico Example Parameters

| Parameter ID | Parameter Name | Type | Default Value |
|---|---|---|---|
| 0x0001 (1) | Oscillator_A. Frequency | Packed_F32.T | (Value=>0.175) |
| 0x0002 (2) | Oscillator_A. Amplitude | Packed_F32.T | (Value=>5.0) |
| 0x0003 (3) | Oscillator_A. Offset | Packed_F32.T | (Value=>0.0) |
| 0x0004 (4) | Oscillator_B. Frequency | Packed_F32.T | (Value=>0.175) |
| 0x0005 (5) | Oscillator_B. Amplitude | Packed_F32.T | (Value=>5.0) |
| 0x0006 (6) | Oscillator_B. Offset | Packed_F32.T | (Value=>0.0) |

Parameter Descriptions:
- **Oscillator_A.Frequency** - The frequency of the oscillator in Hz
- **Oscillator_A.Amplitude** - The amplitude of the oscillator
- **Oscillator_A.Offset** - The Y offset of the oscillator
- **Oscillator_B.Frequency** - The frequency of the oscillator in Hz
- **Oscillator_B.Amplitude** - The amplitude of the oscillator
- **Oscillator_B.Offset** - The Y offset of the oscillator

## 2.7 Events

The table below shows the events for the Pico Example assembly.

Table 6: Pico Example Events

| **Event ID** | **Event Name** | **Parameter Type** |
|---|---|---|
| 0x0001 (1) | Tick_Divider_Instance.<br>Component_Has_Full_Queue | Td_Full_Queue_Param.T |
| 0x0002 (2) | Slow_Rate_Group.Cycle_Slip | Cycle_Slip_Param.T |
| 0x0003 (3) | Slow_Rate_Group.Max_Cycle_<br>Time_Exceeded | Time_Exceeded.T |
| 0x0004 (4) | Slow_Rate_Group.Max_<br>Execution_Time_Exceeded | Time_Exceeded.T |
| 0x0005 (5) | Slow_Rate_Group.Component_<br>Has_Full_Queue | Full_Queue_Param.T |
| 0x0006 (6) | Slow_Rate_Group.Incoming_<br>Tick_Dropped | Tick.T |
| 0x0007 (7) | Fast_Rate_Group.Cycle_Slip | Cycle_Slip_Param.T |
| 0x0008 (8) | Fast_Rate_Group.Max_Cycle_<br>Time_Exceeded | Time_Exceeded.T |
| 0x0009 (9) | Fast_Rate_Group.Max_<br>Execution_Time_Exceeded | Time_Exceeded.T |
| 0x000a (10) | Fast_Rate_Group.Component_<br>Has_Full_Queue | Full_Queue_Param.T |
| 0x000b (11) | Fast_Rate_Group.Incoming_<br>Tick_Dropped | Tick.T |
| 0x000c (12) | Watchdog_Rate_Group.Cycle_<br>Slip | Cycle_Slip_Param.T |
| 0x000d (13) | Watchdog_Rate_Group.Max_<br>Cycle_Time_Exceeded | Time_Exceeded.T |
| 0x000e (14) | Watchdog_Rate_Group.Max_<br>Execution_Time_Exceeded | Time_Exceeded.T |
| 0x000f (15) | Watchdog_Rate_Group.<br>Component_Has_Full_Queue | Full_Queue_Param.T |
| 0x0010 (16) | Watchdog_Rate_Group.<br>Incoming_Tick_Dropped | Tick.T |

| | | |
|---|---|---|
| 0x0011 (17) | Ccsds_Command_Depacketizer_Instance.Invalid_Packet_Checksum | Invalid_Packet_Xor8_Info.T |
| 0x0012 (18) | Ccsds_Command_Depacketizer_Instance.Invalid_Packet_Type | Ccsds_Primary_Header.T |
| 0x0013 (19) | Ccsds_Command_Depacketizer_Instance.Packet_Too_Small | Invalid_Packet_Length.T |
| 0x0014 (20) | Ccsds_Command_Depacketizer_Instance.Packet_Too_Large | Invalid_Packet_Length.T |
| 0x0015 (21) | Ccsds_Command_Depacketizer_Instance.No_Secondary_Header | Ccsds_Primary_Header.T |
| 0x0016 (22) | Ccsds_Command_Depacketizer_Instance.Counts_Reset | - |
| 0x0017 (23) | Ccsds_Command_Depacketizer_Instance.Invalid_Command_Received | Invalid_Command_Info.T |
| 0x0018 (24) | Command_Router_Instance.Command_Received | Command_Header.T |
| 0x0019 (25) | Command_Router_Instance.Command_Execution_Successful | Command_Response.T |
| 0x001a (26) | Command_Router_Instance.Command_Execution_Failure | Command_Response.T |
| 0x001b (27) | Command_Router_Instance.Command_Id_Not_Registered | Command_Header.T |
| 0x001c (28) | Command_Router_Instance.Registration_Id_Conflict | Command_Id.T |
| 0x001d (29) | Command_Router_Instance.Router_Table_Full | Command_Id.T |
| 0x001e (30) | Command_Router_Instance.Outgoing_Command_Dropped | Command_Header.T |
| 0x001f (31) | Command_Router_Instance.Incoming_Command_Dropped | Command_Header.T |

| | | |
|---|---|---|
| 0x0020 (32) | Command_Router_Instance.<br>Noop_Command_Dropped | Command_Header.T |
| 0x0021 (33) | Command_Router_Instance.<br>Command_Response_Dropped | Command_Response.T |
| 0x0022 (34) | Command_Router_Instance.<br>Noop_Received | - |
| 0x0023 (35) | Command_Router_Instance.<br>Noop_Arg_Received | Command_Router_Arg.T |
| 0x0024 (36) | Command_Router_Instance.<br>Noop_Response_Received | - |
| 0x0025 (37) | Command_Router_Instance.<br>Noop_Response_Forwarding_<br>Success | Command_Response.T |
| 0x0026 (38) | Command_Router_Instance.<br>Forwarded_Command_Response_<br>Dropped | Command_Response.T |
| 0x0027 (39) | Command_Router_Instance.<br>Invalid_Command_Source_Id | Command_Response.T |
| 0x0028 (40) | Command_Router_Instance.<br>Invalid_Command_Received | Invalid_Command_Info.T |
| 0x0029 (41) | Command_Router_Instance.<br>Data_Products_Reset | - |
| 0x002a (42) | Event_Filter_Instance.<br>Invalid_Command_Received | Invalid_Command_Info.T |
| 0x002b (43) | Event_Filter_Instance.<br>Filtered_Event | Event_Filter_Single_Event_<br>Cmd_Type.T |
| 0x002c (44) | Event_Filter_Instance.<br>Unfiltered_Event | Event_Filter_Single_Event_<br>Cmd_Type.T |
| 0x002d (45) | Event_Filter_Instance.<br>Filtered_Event_Range | Event_Filter_Id_Range.T |
| 0x002e (46) | Event_Filter_Instance.<br>Unfiltered_Event_Range | Event_Filter_Id_Range.T |
| 0x002f (47) | Event_Filter_Instance.<br>Enable_Event_Filter | - |

| | | |
|---|---|---|
| 0x0030 (48) | Event_Filter_Instance.<br>Disable_Event_Filter | - |
| 0x0031 (49) | Event_Filter_Instance.<br>Filter_Event_Invalid_Id | Event_Filter_Single_Event_<br>Cmd_Type.T |
| 0x0032 (50) | Event_Filter_Instance.<br>Unfilter_Event_Invalid_Id | Event_Filter_Single_Event_<br>Cmd_Type.T |
| 0x0033 (51) | Event_Filter_Instance.<br>Filter_Event_Range_Invalid_<br>Id | Event_Filter_Id_Range.T |
| 0x0034 (52) | Event_Filter_Instance.<br>Unfilter_Event_Range_<br>Invalid_Id | Event_Filter_Id_Range.T |
| 0x0035 (53) | Event_Filter_Instance.Dump_<br>Event_States_Recieved | - |
| 0x0036 (54) | Event_Limiter_Instance.<br>Invalid_Command_Received | Invalid_Command_Info.T |
| 0x0037 (55) | Event_Limiter_Instance.<br>Events_Limited_Since_Last_<br>Tick | Event_Limiter_Num_Events_<br>Type.T |
| 0x0038 (56) | Event_Limiter_Instance.<br>Event_Limit_Enabled | Event_Single_State_Cmd_<br>Type.T |
| 0x0039 (57) | Event_Limiter_Instance.<br>Event_Limit_Disabled | Event_Single_State_Cmd_<br>Type.T |
| 0x003a (58) | Event_Limiter_Instance.<br>Event_Limit_Range_Enabled | Event_Limiter_Id_Range.T |
| 0x003b (59) | Event_Limiter_Instance.<br>Event_Limit_Range_Disabled | Event_Limiter_Id_Range.T |
| 0x003c (60) | Event_Limiter_Instance.<br>Event_Limiting_Enabled | - |
| 0x003d (61) | Event_Limiter_Instance.<br>Event_Limiting_Disabled | - |
| 0x003e (62) | Event_Limiter_Instance.<br>Event_Limit_Enable_Invalid_<br>Id | Event_Single_State_Cmd_<br>Type.T |

| | | |
|---|---|---|
| 0x003f (63) | Event_Limiter_Instance.<br>Event_Limit_Disable_<br>Invalid_Id | Event_Single_State_Cmd_<br>Type.T |
| 0x0040 (64) | Event_Limiter_Instance.<br>Event_Limit_Range_Enabled_<br>Invalid_Id | Event_Limiter_Id_Range.T |
| 0x0041 (65) | Event_Limiter_Instance.<br>Event_Limit_Range_Disabled_<br>Invalid_Id | Event_Limiter_Id_Range.T |
| 0x0042 (66) | Event_Limiter_Instance.Set_<br>New_Persistence | Event_Limiter_Persistence_<br>Type.T |
| 0x0043 (67) | Event_Limiter_Instance.<br>Dump_Event_States_Recieved | - |
| 0x0044 (68) | Product_Database_Instance.<br>Data_Product_Update_Id_Out_<br>Of_Range | Data_Product_Id.T |
| 0x0045 (69) | Product_Database_Instance.<br>Data_Product_Fetch_Id_Out_<br>Of_Range | Data_Product_Id.T |
| 0x0046 (70) | Product_Database_Instance.<br>Data_Product_Fetch_Id_Not_<br>Available | Data_Product_Id.T |
| 0x0047 (71) | Product_Database_Instance.<br>Override_Cleared | Data_Product_Id.T |
| 0x0048 (72) | Product_Database_Instance.<br>Override_Cleared_For_All | - |
| 0x0049 (73) | Product_Database_Instance.<br>Data_Product_Overridden | Data_Product_Header.T |
| 0x004a (74) | Product_Database_Instance.<br>Data_Product_Override_<br>Serialization_Failure | Data_Product_Header.T |
| 0x004b (75) | Product_Database_Instance.<br>Data_Product_Override_Id_<br>Out_Of_Range | Data_Product_Id.T |
| 0x004c (76) | Product_Database_Instance.<br>Data_Product_Clear_<br>Override_Id_Out_Of_Range | Data_Product_Id.T |

| | | |
|---|---|---|
| 0x004d (77) | Product_Database_Instance.Data_Product_Dump_Id_Not_Available | Data_Product_Id.T |
| 0x004e (78) | Product_Database_Instance.Data_Product_Dump_Id_Out_Of_Range | Data_Product_Id.T |
| 0x004f (79) | Product_Database_Instance.Data_Product_Dumped | Data_Product_Header.T |
| 0x0050 (80) | Product_Database_Instance.Dumping_Data_Product_Poly_Type | Data_Product_Poly_Extract.T |
| 0x0051 (81) | Product_Database_Instance.Dumped_Data_Product_Poly_Type | Data_Product_Poly_Event.T |
| 0x0052 (82) | Product_Database_Instance.Data_Product_Dump_Poly_Id_Not_Available | Data_Product_Id.T |
| 0x0053 (83) | Product_Database_Instance.Data_Product_Dump_Poly_Id_Out_Of_Range | Data_Product_Id.T |
| 0x0054 (84) | Product_Database_Instance.Data_Product_Poly_Type_Extraction_Failed | Data_Product_Header.T |
| 0x0055 (85) | Product_Database_Instance.Invalid_Command_Received | Invalid_Command_Info.T |
| 0x0056 (86) | Product_Packetizer_Instance.Invalid_Packet_Id_Commanded | Invalid_Packet_Id.T |
| 0x0057 (87) | Product_Packetizer_Instance.Packet_Enabled | Packet_Period.T |
| 0x0058 (88) | Product_Packetizer_Instance.Packet_Disabled | Packet_Period.T |
| 0x0059 (89) | Product_Packetizer_Instance.Packet_Period_Set | Packet_Period.T |
| 0x005a (90) | Product_Packetizer_Instance.Data_Product_Missing_On_Fetch | Packet_Data_Product_Ids.T |

| | | |
|---|---|---|
| 0x005b (91) | Product_Packetizer_ Instance.Packet_Period_ Item_Bad_Id | Packet_Data_Product_Ids.T |
| 0x005c (92) | Product_Packetizer_ Instance.Data_Product_ Length_Mismatch | Invalid_Data_Product_ Length.T |
| 0x005d (93) | Product_Packetizer_ Instance.Invalid_Command_ Received | Invalid_Command_Info.T |
| 0x005e (94) | Product_Packetizer_ Instance.Dropped_Command | Command_Header.T |
| 0x005f (95) | Ccsds_Serial_Interface_ Instance.Packet_Send_Failed | Ccsds_Primary_Header.T |
| 0x0060 (96) | Ccsds_Serial_Interface_ Instance.Packet_Recv_Failed | Ccsds_Primary_Header.T |
| 0x0061 (97) | Ccsds_Serial_Interface_ Instance.Have_Not_Seen_ Sync_Pattern | Packed_U32.T |
| 0x0062 (98) | Counter_Instance.Set_Count_ Command_Received | Packed_U32.T |
| 0x0063 (99) | Counter_Instance.Reset_ Count_Command_Received | - |
| 0x0064 (100) | Counter_Instance.Set_Count_ Add_Command_Received | Operands.T |
| 0x0065 (101) | Counter_Instance.Sending_ Value | Packed_U32.T |
| 0x0066 (102) | Counter_Instance.Dropped_ Command | Command_Header.T |
| 0x0067 (103) | Counter_Instance.Invalid_ Command_Received | Invalid_Command_Info.T |
| 0x0068 (104) | Oscillator_A.Frequency_ Value_Set | Packed_F32.T |
| 0x0069 (105) | Oscillator_A.Amplitude_ Value_Set | Packed_F32.T |
| 0x006a (106) | Oscillator_A.Offset_Value_ Set | Packed_F32.T |

| | | |
|---|---|---|
| 0x006b (107) | Oscillator_A.Dropped_Command | Command_Header.T |
| 0x006c (108) | Oscillator_A.Invalid_Command_Received | Invalid_Command_Info.T |
| 0x006d (109) | Oscillator_A.Invalid_Parameter_Received | Invalid_Parameter_Info.T |
| 0x006e (110) | Oscillator_B.Frequency_Value_Set | Packed_F32.T |
| 0x006f (111) | Oscillator_B.Amplitude_Value_Set | Packed_F32.T |
| 0x0070 (112) | Oscillator_B.Offset_Value_Set | Packed_F32.T |
| 0x0071 (113) | Oscillator_B.Dropped_Command | Command_Header.T |
| 0x0072 (114) | Oscillator_B.Invalid_Command_Received | Invalid_Command_Info.T |
| 0x0073 (115) | Oscillator_B.Invalid_Parameter_Received | Invalid_Parameter_Info.T |
| 0x0074 (116) | Zero_Divider_Instance.Dividing_By_Zero | Packed_Natural.T |
| 0x0075 (117) | Zero_Divider_Instance.Invalid_Magic_Number | Packed_U32.T |
| 0x0076 (118) | Zero_Divider_Instance.Invalid_Command_Received | Invalid_Command_Info.T |
| 0x0077 (119) | Task_Watchdog_Instance.Watchdog_Pet_Checks_Enabled | - |
| 0x0078 (120) | Task_Watchdog_Instance.Watchdog_Pet_Checks_Disabled | - |
| 0x0079 (121) | Task_Watchdog_Instance.Watchdog_Limit_Set | Watchdog_Limit_Cmd.T |
| 0x007a (122) | Task_Watchdog_Instance.Watchdog_Action_Set | Watchdog_Action_Cmd.T |

| | | |
|---|---|---|
| 0x007b (123) | Task_Watchdog_Instance.<br>Watchdog_Limit_Change_<br>Index_Out_Of_Range | Packed_Connector_Index.T |
| 0x007c (124) | Task_Watchdog_Instance.<br>Watchdog_Action_Change_<br>Index_Out_Of_Range | Packed_Connector_Index.T |
| 0x007d (125) | Task_Watchdog_Instance.<br>Watchdog_Action_Change_<br>Invalid_Transition_To_Fault | Packed_Connector_Index.T |
| 0x007e (126) | Task_Watchdog_Instance.<br>Component_Exceeded_Pet_<br>Limit | Packed_Connector_Index.T |
| 0x007f (127) | Task_Watchdog_Instance.<br>Critical_Task_Not_Petting | Packed_Connector_Index.T |
| 0x0080 (128) | Task_Watchdog_Instance.<br>Invalid_Command_Received | Invalid_Command_Info.T |
| 0x0081 (129) | Fault_Producer_Instance.<br>Sending_Fault_1 | - |
| 0x0082 (130) | Fault_Producer_Instance.<br>Sending_Fault_2 | - |
| 0x0083 (131) | Fault_Producer_Instance.<br>Invalid_Command_Received | Invalid_Command_Info.T |
| 0x0084 (132) | Fault_Correction_Instance.<br>Fault_Received | Fault_Static.T |
| 0x0085 (133) | Fault_Correction_Instance.<br>Fault_Response_Sent | Command_Header.T |
| 0x0086 (134) | Fault_Correction_Instance.<br>Fault_Response_Cleared | Packed_Fault_Id.T |
| 0x0087 (135) | Fault_Correction_Instance.<br>Fault_Response_Disabled | Packed_Fault_Id.T |
| 0x0088 (136) | Fault_Correction_Instance.<br>Fault_Response_Enabled | Packed_Fault_Id.T |
| 0x0089 (137) | Fault_Correction_Instance.<br>All_Fault_Responses_Cleared | - |
| 0x008a (138) | Fault_Correction_Instance.<br>Unrecognized_Fault_Id | Packed_Fault_Id.T |

| | | |
|---|---|---|
| 0x008b (139) | Fault_Correction_Instance.<br>Invalid_Command_Received | Invalid_Command_Info.T |
| 0x008c (140) | Fault_Correction_Instance.<br>Command_Dropped | Command_Header.T |
| 0x008d (141) | Fault_Correction_Instance.<br>Fault_Dropped | Fault_Header.T |
| 0x008e (142) | Fault_Correction_Instance.<br>Data_Products_Reset | - |
| 0x008f (143) | Cpu_Monitor_Instance.<br>Packet_Period_Set | Packed_U16.T |
| 0x0090 (144) | Cpu_Monitor_Instance.<br>Invalid_Command_Received | Invalid_Command_Info.T |
| 0x0091 (145) | Queue_Monitor_Instance.<br>Packet_Period_Set | Packed_U16.T |
| 0x0092 (146) | Queue_Monitor_Instance.<br>Invalid_Command_Received | Invalid_Command_Info.T |
| 0x0093 (147) | Stack_Monitor_Instance.<br>Packet_Period_Set | Packed_U16.T |
| 0x0094 (148) | Stack_Monitor_Instance.<br>Invalid_Command_Received | Invalid_Command_Info.T |

Event Descriptions:

- **Tick_Divider_Instance.Component_Has_Full_Queue** - The tick divider tried to put a Tick on a component's queue, but the queue was full, so the Tick was dropped.
- **Slow_Rate_Group.Cycle_Slip** - Execution ran long on this cycle.
- **Slow_Rate_Group.Max_Cycle_Time_Exceeded** - A new maximum cycle time was reached. The event parameter is a Tick type with the maximum cycle time as the Time and the cycle count where the maximum cycle was achieved as the Count.
- **Slow_Rate_Group.Max_Execution_Time_Exceeded** - A new maximum execution time was reached. The event parameter is a Tick type with the maximum cycle time as the Time and the cycle count where the maximum cycle was achieved as the Count.
- **Slow_Rate_Group.Component_Has_Full_Queue** - The rate group tried to put a Tick on a component's queue, but the queue was full, so the Tick was dropped.
- **Slow_Rate_Group.Incoming_Tick_Dropped** - The rate group component's queue is full, so it cannot store the tick coming in. This usually means the rate group is cycle slipping and not running as fast as it needs to.
- **Fast_Rate_Group.Cycle_Slip** - Execution ran long on this cycle.
- **Fast_Rate_Group.Max_Cycle_Time_Exceeded** - A new maximum cycle time was reached. The event parameter is a Tick type with the maximum cycle time as the Time and the cycle count where the maximum cycle was achieved as the Count.

- **Fast_Rate_Group.Max_Execution_Time_Exceeded** - A new maximum execution time was reached. The event parameter is a Tick type with the maximum cycle time as the Time and the cycle count where the maximum cycle was achieved as the Count.
- **Fast_Rate_Group.Component_Has_Full_Queue** - The rate group tried to put a Tick on a component's queue, but the queue was full, so the Tick was dropped.
- **Fast_Rate_Group.Incoming_Tick_Dropped** - The rate group component's queue is full, so it cannot store the tick coming in. This usually means the rate group is cycle slipping and not running as fast as it needs to.
- **Watchdog_Rate_Group.Cycle_Slip** - Execution ran long on this cycle.
- **Watchdog_Rate_Group.Max_Cycle_Time_Exceeded** - A new maximum cycle time was reached. The event parameter is a Tick type with the maximum cycle time as the Time and the cycle count where the maximum cycle was achieved as the Count.
- **Watchdog_Rate_Group.Max_Execution_Time_Exceeded** - A new maximum execution time was reached. The event parameter is a Tick type with the maximum cycle time as the Time and the cycle count where the maximum cycle was achieved as the Count.
- **Watchdog_Rate_Group.Component_Has_Full_Queue** - The rate group tried to put a Tick on a component's queue, but the queue was full, so the Tick was dropped.
- **Watchdog_Rate_Group.Incoming_Tick_Dropped** - The rate group component's queue is full, so it cannot store the tick coming in. This usually means the rate group is cycle slipping and not running as fast as it needs to.
- **Ccsds_Command_Depacketizer_Instance.Invalid_Packet_Checksum** - A packet was received with an invalid checksum
- **Ccsds_Command_Depacketizer_Instance.Invalid_Packet_Type** - A packet was received with an invalid ccsds packet type. The expected packet type is a telecommand, but a telemtry packet was received.
- **Ccsds_Command_Depacketizer_Instance.Packet_Too_Small** - The packet recieved was too small to contain necessary command information.
- **Ccsds_Command_Depacketizer_Instance.Packet_Too_Large** - The packet recieved was too large and is bigger than the size of a command.
- **Ccsds_Command_Depacketizer_Instance.No_Secondary_Header** - A packet was received without a secondary header, but the secondary header is required.
- **Ccsds_Command_Depacketizer_Instance.Counts_Reset** - A command was received to reset the counts.
- **Ccsds_Command_Depacketizer_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Command_Router_Instance.Command_Received** - A command was received by the command router to be routed.
- **Command_Router_Instance.Command_Execution_Successful** - A command was routed, executed, and returned a response saying it was executed successfully
- **Command_Router_Instance.Command_Execution_Failure** - A command execution failed.
- **Command_Router_Instance.Command_Id_Not_Registered** - A command was sent to the router, but it was not found in the router table.
- **Command_Router_Instance.Registration_Id_Conflict** - The command Id has already been registered.
- **Command_Router_Instance.Router_Table_Full** - Cannot add command Id to router table because it is full.
- **Command_Router_Instance.Outgoing_Command_Dropped** - A command was dropped

because the recipient's queue was full.

- **Command_Router_Instance.Incoming_Command_Dropped** - A command was dropped because the command router's queue was full.
- **Command_Router_Instance.Noop_Command_Dropped** - A noop command was dropped because the command router's queue was full.
- **Command_Router_Instance.Command_Response_Dropped** - A command response was dropped because the command router's queue was full.
- **Command_Router_Instance.Noop_Received** - A Noop command was received.
- **Command_Router_Instance.Noop_Arg_Received** - A Noop command was received with an argument.
- **Command_Router_Instance.Noop_Response_Received** - A noop response self test command was received.
- **Command_Router_Instance.Noop_Response_Forwarding_Success** - If this event is sent then the noop response self test command succeeded.
- **Command_Router_Instance.Forwarded_Command_Response_Dropped** - A forwarded command response was dropped because the receiving component's queue overflowed.
- **Command_Router_Instance.Invalid_Command_Source_Id** - A command response contained an invalid source id. This is a software bug and should be corrected.
- **Command_Router_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Command_Router_Instance.Data_Products_Reset** - The component's data products have been reset to initialization values.
- **Event_Filter_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Event_Filter_Instance.Filtered_Event** - This event indicates that the state of an event was set to enabled for the filter.
- **Event_Filter_Instance.Unfiltered_Event** - This event indicates that the state of an event was set to disabled for the filter.
- **Event_Filter_Instance.Filtered_Event_Range** - This event indicates that the state of a range of events were set to enabled for the filter.
- **Event_Filter_Instance.Unfiltered_Event_Range** - This event indicates that the state of a range of events were set to disabled for the filter.
- **Event_Filter_Instance.Enable_Event_Filter** - This event indicates that the state of all events were set to enabled for the filter, but kept the internal state.
- **Event_Filter_Instance.Disable_Event_Filter** - This event indicates that the state of all events were set to disabled for the filter, but kept the internal state.
- **Event_Filter_Instance.Filter_Event_Invalid_Id** - This event indicates that the command to change the event state to enabled failed since the event ID was out of range.
- **Event_Filter_Instance.Unfilter_Event_Invalid_Id** - This event indicates that the command to change the event state to disable failed since the event ID was out of range.
- **Event_Filter_Instance.Filter_Event_Range_Invalid_Id** - This event indicates that changing the state for the range to enabled, failed due to an invalid id.
- **Event_Filter_Instance.Unfilter_Event_Range_Invalid_Id** - This event indicates that changing the state for the range to disabled, failed due to an invalid id.
- **Event_Filter_Instance.Dump_Event_States_Recieved** - Event that indicates the process of building the packet that stores the event states has started and will send the packet once we go through a decrement cycle.

- **Event_Limiter_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Event_Limiter_Instance.Events_Limited_Since_Last_Tick** - An event that indicates how many events have been limited as well as up to the first 10 ids of those events. The event ids listed may not have been dropped, but are at listed since they are at the current max limit.
- **Event_Limiter_Instance.Event_Limit_Enabled** - This event indicates that the state of an event was set to enabled for the limiter.
- **Event_Limiter_Instance.Event_Limit_Disabled** - This event indicates that the state of an event was set to disabled for the limiter.
- **Event_Limiter_Instance.Event_Limit_Range_Enabled** - This event indicates that the state of a range of events were set to enabled for the limiter.
- **Event_Limiter_Instance.Event_Limit_Range_Disabled** - This event indicates that the state of a range of events were set to disabled for the limiter.
- **Event_Limiter_Instance.Event_Limiting_Enabled** - This event indicates that the state of all events were set to enabled for the limiter.
- **Event_Limiter_Instance.Event_Limiting_Disabled** - This event indicates that the state of all events were set to disabled for the limiter.
- **Event_Limiter_Instance.Event_Limit_Enable_Invalid_Id** - This event indicates that the command to change the event state to enabled failed since the event ID was out of range.
- **Event_Limiter_Instance.Event_Limit_Disable_Invalid_Id** - This event indicates that the command to change the event state to disable failed since the event ID was out of range.
- **Event_Limiter_Instance.Event_Limit_Range_Enabled_Invalid_Id** - This event indicates that changing the state for the range to enabled, failed due to an invalid id.
- **Event_Limiter_Instance.Event_Limit_Range_Disabled_Invalid_Id** - This event indicates that changing the state for the range to disabled, failed due to an invalid id.
- **Event_Limiter_Instance.Set_New_Persistence** - Indicates that the persistence of the number of events until we limit was changed to a new value between 1 and 7.
- **Event_Limiter_Instance.Dump_Event_States_Recieved** - Event that indicates the process of building the packet that stores the event states has started and will send the packet once we go through a decrement cycle.
- **Product_Database_Instance.Data_Product_Update_Id_Out_Of_Range** - A data product update was received with an ID that was out of range.
- **Product_Database_Instance.Data_Product_Fetch_Id_Out_Of_Range** - A data product fetch was received with an ID that was out of range.
- **Product_Database_Instance.Data_Product_Fetch_Id_Not_Available** - A data product fetch was received with an ID that has not yet been stored in the database.
- **Product_Database_Instance.Override_Cleared** - Override condition cleared for the data product of the provided ID.
- **Product_Database_Instance.Override_Cleared_For_All** - Override condition cleared for all data productd.
- **Product_Database_Instance.Data_Product_Overridden** - Data product overridden by command.
- **Product_Database_Instance.Data_Product_Override_Serialization_Failure** - Data product override could not be completed due to a serialization error.
- **Product_Database_Instance.Data_Product_Override_Id_Out_Of_Range** - A

data product override command was received with an ID that was out of range.

- **Product_Database_Instance.Data_Product_Clear_Override_Id_Out_Of_Range** - A data product clear override command was received with an ID that was out of range.
- **Product_Database_Instance.Data_Product_Dump_Id_Not_Available** - A data product dump command was received with an ID that has not yet been stored in the database.
- **Product_Database_Instance.Data_Product_Dump_Id_Out_Of_Range** - A data product dump command was received with an ID that was out of range.
- **Product_Database_Instance.Data_Product_Dumped** - Data product dumped into a packet by command.
- **Product_Database_Instance.Dumping_Data_Product_Poly_Type** - Data product poly type dumped into a packet by command.
- **Product_Database_Instance.Dumped_Data_Product_Poly_Type** - Data product poly type dumped into a packet by command.
- **Product_Database_Instance.Data_Product_Dump_Poly_Id_Not_Available** - A data product dump poly command was received with an ID that has not yet been stored in the database.
- **Product_Database_Instance.Data_Product_Dump_Poly_Id_Out_Of_Range** - A data product dump poly command was received with an ID that was out of range.
- **Product_Database_Instance.Data_Product_Poly_Type_Extraction_Failed** - A data product dump poly command failed because the extraction could not succeed with the provided parameters.
- **Product_Database_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Product_Packetizer_Instance.Invalid_Packet_Id_Commanded** - An invalid packet id was commanded for a given command.
- **Product_Packetizer_Instance.Packet_Enabled** - An packet was enabled.
- **Product_Packetizer_Instance.Packet_Disabled** - An packet was disabled.
- **Product_Packetizer_Instance.Packet_Period_Set** - An packet period was set.
- **Product_Packetizer_Instance.Data_Product_Missing_On_Fetch** - A data product was missing when fetched for packet insertion.
- **Product_Packetizer_Instance.Packet_Period_Item_Bad_Id** - A packet period packet item could not be formed because the ID is invalid.
- **Product_Packetizer_Instance.Data_Product_Length_Mismatch** - A data product was fetched but contained an unexpected length.
- **Product_Packetizer_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Product_Packetizer_Instance.Dropped_Command** - A command was dropped due to a full queue.
- **Ccsds_Serial_Interface_Instance.Packet_Send_Failed** - Failed to send a packet over the socket because it has an invalid CCSDS header.
- **Ccsds_Serial_Interface_Instance.Packet_Recv_Failed** - Failed to receive a packet over the socket because it has an invalid CCSDS header.
- **Ccsds_Serial_Interface_Instance.Have_Not_Seen_Sync_Pattern** - The component as received N number of bytes without seeing a sync pattern yet.
- **Counter_Instance.Set_Count_Command_Received** - Received a Set_Count command.
- **Counter_Instance.Reset_Count_Command_Received** - Received a Reset_Count com-

mand.

- **Counter_Instance.Set_Count_Add_Command_Received** - Received a Set_Count_Add command.
- **Counter_Instance.Sending_Value** - Sending the current value out as data product.
- **Counter_Instance.Dropped_Command** - The component's queue overflowed and the command was dropped.
- **Counter_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Oscillator_A.Frequency_Value_Set** - A new frequency value was set by command
- **Oscillator_A.Amplitude_Value_Set** - A new amplitude value was set by command
- **Oscillator_A.Offset_Value_Set** - A new offset value was set by command
- **Oscillator_A.Dropped_Command** - The component's queue overflowed and the command was dropped.
- **Oscillator_A.Invalid_Command_Received** - A command was received with invalid parameters.
- **Oscillator_A.Invalid_Parameter_Received** - A parameter was received with invalid parameters.
- **Oscillator_B.Frequency_Value_Set** - A new frequency value was set by command
- **Oscillator_B.Amplitude_Value_Set** - A new amplitude value was set by command
- **Oscillator_B.Offset_Value_Set** - A new offset value was set by command
- **Oscillator_B.Dropped_Command** - The component's queue overflowed and the command was dropped.
- **Oscillator_B.Invalid_Command_Received** - A command was received with invalid parameters.
- **Oscillator_B.Invalid_Parameter_Received** - A parameter was received with invalid parameters.
- **Zero_Divider_Instance.Dividing_By_Zero** - A divide by zero command was received, and the magic number was correct. The division will occur in N milliseconds, where N is provided as the event parameter.
- **Zero_Divider_Instance.Invalid_Magic_Number** - A divide by zero command was received, but the magic number was incorrect. The division will not occur.
- **Zero_Divider_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Task_Watchdog_Instance.Watchdog_Pet_Checks_Enabled** - Indicates a command was recieved to enable the checks on upstream pets.
- **Task_Watchdog_Instance.Watchdog_Pet_Checks_Disabled** - Indicates a command was recieved to disable the checks on upstream pets.
- **Task_Watchdog_Instance.Watchdog_Limit_Set** - An event to indicate that the limit was changed by command for a particular index.
- **Task_Watchdog_Instance.Watchdog_Action_Set** - An event to indicate that the action was changed by command for a particular index.
- **Task_Watchdog_Instance.Watchdog_Limit_Change_Index_Out_Of_Range** - Event indicating there was an error for the index range in the set limit command.
- **Task_Watchdog_Instance.Watchdog_Action_Change_Index_Out_Of_Range** - Event indicating there was an error for the index range in the set limit command.
- **Task_Watchdog_Instance.Watchdog_Action_Change_Invalid_Transition_To_**

**Fault** - Event indicating there was an error trying to set the action to fault. The petter did not have a fault declared in the model so the action cannot be set to fault.

- **Task_Watchdog_Instance.Component_Exceeded_Pet_Limit** - Event to indicate a pet connector has not received a pet within the set limits for that component.
- **Task_Watchdog_Instance.Critical_Task_Not_Petting** - Event to indicate that one or more of our critical tasks have not indicated a pet in the maximum limit of ticks. The hardware watchdog will not be pet in this case.
- **Task_Watchdog_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Fault_Producer_Instance.Sending_Fault_1** - The component received a command to send out fault 1.
- **Fault_Producer_Instance.Sending_Fault_2** - The component received a command to send out fault 2.
- **Fault_Producer_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Fault_Correction_Instance.Fault_Received** - A fault was received.
- **Fault_Correction_Instance.Fault_Response_Sent** - A fault response was sent with the included command header.
- **Fault_Correction_Instance.Fault_Response_Cleared** - A fault response was cleared.
- **Fault_Correction_Instance.Fault_Response_Disabled** - A fault response has been disabled
- **Fault_Correction_Instance.Fault_Response_Enabled** - A fault response has been enabled.
- **Fault_Correction_Instance.All_Fault_Responses_Cleared** - Any latched faults have been unlatched by command.
- **Fault_Correction_Instance.Unrecognized_Fault_Id** - A fault response entry with the included fault ID was not found in the table.
- **Fault_Correction_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Fault_Correction_Instance.Command_Dropped** - A command was dropped due to a full queue.
- **Fault_Correction_Instance.Fault_Dropped** - A fault was dropped due to a full queue.
- **Fault_Correction_Instance.Data_Products_Reset** - The component's data products have been reset to initialization values.
- **Cpu_Monitor_Instance.Packet_Period_Set** - A command was received to change the packet period.
- **Cpu_Monitor_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Queue_Monitor_Instance.Packet_Period_Set** - A command was received to change the packet period.
- **Queue_Monitor_Instance.Invalid_Command_Received** - A command was received with invalid parameters.
- **Stack_Monitor_Instance.Packet_Period_Set** - A command was received to change the packet period.
- **Stack_Monitor_Instance.Invalid_Command_Received** - A command was received with invalid parameters.

## 2.8 Data Products

The table below shows the data products for the Pico Example assembly.

Table 7: Pico Example Data Products

| Data Product ID | Data Product Name | Type |
|---|---|---|
| 0x0001 (1) | Slow_Rate_Group.Timing_Report | Task_Timing_Report.T |
| 0x0002 (2) | Fast_Rate_Group.Timing_Report | Task_Timing_Report.T |
| 0x0003 (3) | Watchdog_Rate_Group.Timing_Report | Task_Timing_Report.T |
| 0x0004 (4) | Ccsds_Command_Depacketizer_Instance.Rejected_Packet_Count | Packed_U16.T |
| 0x0005 (5) | Ccsds_Command_Depacketizer_Instance.Accepted_Packet_Count | Packed_U16.T |
| 0x0006 (6) | Command_Router_Instance.Command_Receive_Count | Packed_U16.T |
| 0x0007 (7) | Command_Router_Instance.Command_Success_Count | Packed_U16.T |
| 0x0008 (8) | Command_Router_Instance.Command_Failure_Count | Packed_U16.T |
| 0x0009 (9) | Command_Router_Instance.Last_Received_Command | Command_Id.T |
| 0x000a (10) | Command_Router_Instance.Last_Successful_Command | Command_Id.T |
| 0x000b (11) | Command_Router_Instance.Last_Failed_Command | Command_Id_Status.T |
| 0x000c (12) | Command_Router_Instance.Noop_Arg_Last_Value | Command_Router_Arg.T |
| 0x000d (13) | Event_Filter_Instance.Total_Events_Filtered | Packed_U32.T |
| 0x000e (14) | Event_Filter_Instance.Total_Events_Unfiltered | Packed_U32.T |

| | | |
|---|---|---|
| 0x000f (15) | Event_Filter_Instance.<br>Component_Filter_State | Event_Component_State_<br>Type.T |
| 0x0010 (16) | Event_Limiter_Instance.<br>Limited_Events_Since_Tick | Packed_U16.T |
| 0x0011 (17) | Event_Limiter_Instance.<br>Total_Events_Limited | Packed_U32.T |
| 0x0012 (18) | Event_Limiter_Instance.<br>Component_Limiting_<br>Enabled_Status | Event_Enable_State_Type.T |
| 0x0013 (19) | Event_Packetizer_Instance.<br>Events_Dropped_Count | Packed_U32.T |
| 0x0014 (20) | Event_Packetizer_Instance.<br>Bytes_Available | Packed_Natural.T |
| 0x0015 (21) | Product_Database_Instance.<br>Data_Product_Poly_Type_<br>Dump | Data_Product_Poly_Type.T |
| 0x0016 (22) | Product_Database_Instance.<br>Database_Override | Packed_Enable_Disable_<br>Type.T |
| 0x0017 (23) | Oscillator_A.Oscillator_<br>Value | Packed_F32.T |
| 0x0018 (24) | Oscillator_B.Oscillator_<br>Value | Packed_F32.T |
| 0x0019 (25) | Adc_Data_Collector_<br>Instance.Channel_0 | Packed_Integer.T |
| 0x001a (26) | Adc_Data_Collector_<br>Instance.Vsys | Packed_Integer.T |
| 0x001b (27) | Adc_Data_Collector_<br>Instance.Temperature | Packed_Integer.T |
| 0x001c (28) | Task_Watchdog_Instance.<br>Watchdog_Component_Petter_<br>State | Packed_Watchdog_Component_<br>State.T |
| 0x001d (29) | Task_Watchdog_Instance.<br>Pet_Connector_Action_<br>States | Pico_Example_Task_<br>Watchdog_List_State_<br>Record.T |
| 0x001e (30) | Task_Watchdog_Instance.<br>Slow_Rate_Group_Limit | Packed_Missed_Pet_Limit.T |

| | | |
|---|---|---|
| 0x001f (31) | Task_Watchdog_Instance.<br>Fast_Rate_Group_Limit | Packed_Missed_Pet_Limit.T |
| 0x0020 (32) | Fault_Correction_Instance.<br>Fault_Counter | Packed_U16.T |
| 0x0021 (33) | Fault_Correction_Instance.<br>Last_Fault_Id_Received | Pico_Example_Fault_<br>Responses_Packed_Id_Type.T |
| 0x0022 (34) | Fault_Correction_Instance.<br>Time_Of_Last_Fault_<br>Received | Sys_Time.T |
| 0x0023 (35) | Fault_Correction_Instance.<br>Fault_Response_Statuses | Pico_Example_Fault_<br>Responses_Status_Record.T |
| 0x0024 (36) | Cpu_Monitor_Instance.<br>Packet_Period | Packed_U16.T |
| 0x0025 (37) | Queue_Monitor_Instance.<br>Packet_Period | Packed_U16.T |
| 0x0026 (38) | Stack_Monitor_Instance.<br>Packet_Period | Packed_U16.T |

Data Product Descriptions:

- **Slow_Rate_Group.Timing_Report** - Data relating timing performance of the component.
- **Fast_Rate_Group.Timing_Report** - Data relating timing performance of the component.
- **Watchdog_Rate_Group.Timing_Report** - Data relating timing performance of the component.
- **Ccsds_Command_Depacketizer_Instance.Rejected_Packet_Count** - The number of packets rejected by the component due to invalid data
- **Ccsds_Command_Depacketizer_Instance.Accepted_Packet_Count** - The number of packets accepted by the component
- **Command_Router_Instance.Command_Receive_Count** - The number of commands received by the component.
- **Command_Router_Instance.Command_Success_Count** - The number of commands that successfully executed.
- **Command_Router_Instance.Command_Failure_Count** - The number of commands that failed to execute.
- **Command_Router_Instance.Last_Received_Command** - The ID of the last received command by the command router.
- **Command_Router_Instance.Last_Successful_Command** - The ID of the last successful command routed by the command router.
- **Command_Router_Instance.Last_Failed_Command** - The ID and status of the last failed command routed by the command router.
- **Command_Router_Instance.Noop_Arg_Last_Value** - The last value sent with the Noop_Arg command. This data product can be useful for testing purposes.

- **Event_Filter_Instance.Total_Events_Filtered** - The total number of events that have been filtered for the components lifetime.
- **Event_Filter_Instance.Total_Events_Unfiltered** - The total number of events that have been passed through for the components lifetime. Does not include out of range IDs.
- **Event_Filter_Instance.Component_Filter_State** - The state of the master switch for filtering events.
- **Event_Limiter_Instance.Limited_Events_Since_Tick** - The number of events that were limited since the last tick.
- **Event_Limiter_Instance.Total_Events_Limited** - The total number of events that have been limited for the components lifetime.
- **Event_Limiter_Instance.Component_Limiting_Enabled_Status** - The current state of the component master switch.
- **Event_Packetizer_Instance.Events_Dropped_Count** - The number of events dropped by the component.
- **Event_Packetizer_Instance.Bytes_Available** - The current number of bytes available for event storage within the component.
- **Product_Database_Instance.Data_Product_Poly_Type_Dump** - Data product poly type dumped into a data product by command.
- **Product_Database_Instance.Database_Override** - If set to Enabled then the database contains at least one data product that has been overridden by command.
- **Oscillator_A.Oscillator_Value** - The current value of the oscillator.
- **Oscillator_B.Oscillator_Value** - The current value of the oscillator.
- **Adc_Data_Collector_Instance.Channel_0** - The ADC reading at Channel 0 in microvolts.
- **Adc_Data_Collector_Instance.Vsys** - The ADC reading of the system voltage in microvolts.
- **Adc_Data_Collector_Instance.Temperature** - The ADC temperature reading in Celsius.
- **Task_Watchdog_Instance.Watchdog_Component_Petter_State** - Data product that tracks the global state to enable or disable all checks on the upstream watchdog pets.
- **Task_Watchdog_Instance.Pet_Connector_Action_States** - 2-bit of state for each pet connector indicating the current action that will be taken if there is an error. Note that Packed_U32.T is just a placeholder type for this data product. The actual type of this data product will be autocoded and at assembly model ingest time.
- **Task_Watchdog_Instance.Slow_Rate_Group_Limit** - Slow rate group monitoring.
- **Task_Watchdog_Instance.Fast_Rate_Group_Limit** - Fast rate group monitoring.
- **Fault_Correction_Instance.Fault_Counter** - The number of faults received by the component.
- **Fault_Correction_Instance.Last_Fault_Id_Received** - The ID of the last fault received.
- **Fault_Correction_Instance.Time_Of_Last_Fault_Received** - The system time of the last fault received.
- **Fault_Correction_Instance.Fault_Response_Statuses** - 2-bits of status for each fault response that this component is managing. Note that Packed_U32.T is just a placeholder type for this data product. The actual type of this data product will be autocoded and at assembly model ingest time.
- **Cpu_Monitor_Instance.Packet_Period** - The current packet period.

- **Queue_Monitor_Instance.Packet_Period** - The current packet period.
- **Stack_Monitor_Instance.Packet_Period** - The current packet period.

## 2.9 Packets

The table below shows the packets for the Pico Example assembly.

Table 8: Pico Example Packets

| Packet ID | Packet Name | Type |
|---|---|---|
| 0x0001 (1) | Product_Packetizer_ Instance.Housekeeping_ Packet | *Undefined* |
| 0x0002 (2) | Ccsds_Command_Depacketizer_ Instance.Error_Packet | Ccsds_Space_Packet.T |
| 0x0003 (3) | Event_Filter_Instance. Event_Filter_State_Packet | *Undefined* |
| 0x0004 (4) | Event_Limiter_Instance. Event_Limiter_State_Packet | *Undefined* |
| 0x0005 (5) | Product_Database_Instance. Dump_Packet | Data_Product.T |
| 0x0006 (6) | Cpu_Monitor_Instance.Cpu_ Usage_Packet | Pico_Example_Cpu_Monitor_ Packet_Type.T |
| 0x0007 (7) | Counter_Instance.Counter_ Value | Packed_U32.T |
| 0x0008 (8) | Queue_Monitor_Instance. Queue_Usage_Packet | Pico_Example_Queue_Monitor_ Packet_Type.T |
| 0x0009 (9) | Stack_Monitor_Instance. Stack_Usage_Packet | Pico_Example_Stack_Monitor_ Packet_Type.T |
| 0x0061 (97) | Zero_Divider_Instance.Last_ Chance_Handler_Packet | Packed_Exception_ Occurrence.T |
| 0x0062 (98) | Event_Packetizer_Instance. Events_Packet | *Undefined* |

Packet Descriptions:

- **Product_Packetizer_Instance.Housekeeping_Packet** - This packet contains housekeeping data.
- **Ccsds_Command_Depacketizer_Instance.Error_Packet** - This packet contains a CCSDS packet that was dropped due to error.

- **Event_Filter_Instance.Event_Filter_State_Packet** - The packet used to dump all the state information for which events are filtered and which are not. Each event ID takes a bit and any extra bits beyond the event range will show as not filtered.
- **Event_Limiter_Instance.Event_Limiter_State_Packet** - The packet used to dump all the state information for which events are limited and which are not. Each event takes a bit and any extra bits beyond the event range will show as disabled.
- **Product_Database_Instance.Dump_Packet** - This packet contains dumped data products.
- **Cpu_Monitor_Instance.Cpu_Usage_Packet** - This packet contains cpu usage numbers for tasks and interrupts in the system.
- **Counter_Instance.Counter_Value** - The counter value 1.
- **Queue_Monitor_Instance.Queue_Usage_Packet** - This packet contains queue usage numbers for queued components in the system.
- **Stack_Monitor_Instance.Stack_Usage_Packet** - This packet contains stack and secondary stack usage numbers for tasks in the system.
- **Zero_Divider_Instance.Last_Chance_Handler_Packet** - This packet contains information regarding an exception occurrence that triggers the Last_Chance_Handler to get invoked. This packet is not produced directly by this component, and should be produced by the last chance handler implementation. This packet definition exists to ensure that the packet gets reflected in the documentation and ground system definitions.
- **Event_Packetizer_Instance.Events_Packet** - This packet contains events as subpackets.

## 2.10 Faults

The table below shows the faults for the Pico Example assembly.

Table 9: Pico Example Faults

| Fault ID | Fault Name | Parameter Type |
|---|---|---|
| 0x0001 (1) | Task_Watchdog_Instance.Slow_Rate_Group_Fault | Packed_Connector_Index.T |
| 0x0002 (2) | Task_Watchdog_Instance.Fast_Rate_Group_Fault | Packed_Connector_Index.T |
| 0x0003 (3) | Fault_Producer_Instance.Fault_1 | - |
| 0x0004 (4) | Fault_Producer_Instance.Fault_2 | Packed_Natural.T |

Fault Descriptions:

- **Task_Watchdog_Instance.Slow_Rate_Group_Fault** - Slow rate group monitoring.
- **Task_Watchdog_Instance.Fast_Rate_Group_Fault** - Fast rate group monitoring.
- **Fault_Producer_Instance.Fault_1** - First fault that the component can send.
- **Fault_Producer_Instance.Fault_2** - Second fault that the component can send.

# 3 Appendix

## 3.1 Connections

Table 10: Pico Example Connections

| Number | From | To | Kind |
|---|---|---|---|
| 1 | Ticker_Instance.Sys_Time_T_Get | System_Time_Instance.Sys_Time_T_Return | get-return |
| 2 | Tick_Divider_Instance.Sys_Time_T_Get | System_Time_Instance.Sys_Time_T_Return | get-return |
| 3 | Slow_Rate_Group.Sys_Time_T_Get | System_Time_Instance.Sys_Time_T_Return | get-return |
| 4 | Fast_Rate_Group.Sys_Time_T_Get | System_Time_Instance.Sys_Time_T_Return | get-return |
| 5 | Watchdog_Rate_Group.Sys_Time_T_Get | System_Time_Instance.Sys_Time_T_Return | get-return |
| 6 | Command_Router_Instance.Sys_Time_T_Get | System_Time_Instance.Sys_Time_T_Return | get-return |
| 7 | Counter_Instance.Sys_Time_T_Get | System_Time_Instance.Sys_Time_T_Return | get-return |
| 8 | Oscillator_A.Sys_Time_T_Get | System_Time_Instance.Sys_Time_T_Return | get-return |
| 9 | Oscillator_B.Sys_Time_T_Get | System_Time_Instance.Sys_Time_T_Return | get-return |
| 10 | Event_Packetizer_Instance.Sys_Time_T_Get | System_Time_Instance.Sys_Time_T_Return | get-return |
| 11 | Product_Database_Instance.Sys_Time_T_Get | System_Time_Instance.Sys_Time_T_Return | get-return |
| 12 | Product_Packetizer_Instance.Sys_Time_T_Get | System_Time_Instance.Sys_Time_T_Return | get-return |

| | | | |
|---|---|---|---|
| 13 | Ccsds_Command_ Depacketizer_ Instance.Sys_Time_ T_Get | System_Time_Instance. Sys_Time_T_Return | get-return |
| 14 | Cpu_Monitor_Instance. Sys_Time_T_Get | System_Time_Instance. Sys_Time_T_Return | get-return |
| 15 | Stack_Monitor_ Instance.Sys_Time_ T_Get | System_Time_Instance. Sys_Time_T_Return | get-return |
| 16 | Queue_Monitor_ Instance.Sys_Time_ T_Get | System_Time_Instance. Sys_Time_T_Return | get-return |
| 17 | Event_Filter_ Instance.Sys_Time_ T_Get | System_Time_Instance. Sys_Time_T_Return | get-return |
| 18 | Event_Limiter_ Instance.Sys_Time_ T_Get | System_Time_Instance. Sys_Time_T_Return | get-return |
| 19 | Zero_Divider_ Instance.Sys_Time_ T_Get | System_Time_Instance. Sys_Time_T_Return | get-return |
| 20 | Task_Watchdog_ Instance.Sys_Time_ T_Get | System_Time_Instance. Sys_Time_T_Return | get-return |
| 21 | Fault_Correction_ Instance.Sys_Time_T_ Get | System_Time_Instance. Sys_Time_T_Return | get-return |
| 22 | Fault_Producer_ Instance.Sys_Time_ T_Get | System_Time_Instance. Sys_Time_T_Return | get-return |
| 23 | Ccsds_Serial_ Interface_Instance. Sys_Time_T_Get | System_Time_Instance. Sys_Time_T_Return | get-return |
| 24 | Adc_Data_Collector_ Instance.Sys_Time_T_ Get | System_Time_Instance. Sys_Time_T_Return | get-return |
| 25 | Ticker_Instance.Tick_ T_Send | Tick_Divider_ Instance.Tick_T_Recv_ Sync | send-recv_sync |

| | | | |
|---|---|---|---|
| 26 | Tick_Divider_ Instance.Tick_T_Send [1] | Watchdog_Rate_Group. Tick_T_Recv_Async | send-recv_async |
| 27 | Tick_Divider_ Instance.Tick_T_Send [2] | Slow_Rate_Group.Tick_ T_Recv_Async | send-recv_async |
| 28 | Tick_Divider_ Instance.Tick_T_Send [3] | Fast_Rate_Group.Tick_ T_Recv_Async | send-recv_async |
| 29 | Slow_Rate_Group.Tick_ T_Send [1] | Counter_Instance. Tick_T_Recv_Sync | send-recv_sync |
| 30 | Slow_Rate_Group.Tick_ T_Send [2] | Event_Packetizer_ Instance.Tick_T_Recv_ Sync | send-recv_sync |
| 31 | Slow_Rate_Group.Tick_ T_Send [3] | Cpu_Monitor_Instance. Tick_T_Recv_Sync | send-recv_sync |
| 32 | Slow_Rate_Group.Tick_ T_Send [4] | Queue_Monitor_ Instance.Tick_T_Recv_ Sync | send-recv_sync |
| 33 | Slow_Rate_Group.Tick_ T_Send [5] | Stack_Monitor_ Instance.Tick_T_Recv_ Sync | send-recv_sync |
| 34 | Slow_Rate_Group.Tick_ T_Send [6] | Event_Filter_ Instance.Tick_T_Recv_ Sync | send-recv_sync |
| 35 | Slow_Rate_Group.Tick_ T_Send [7] | Event_Limiter_ Instance.Tick_T_Recv_ Sync | send-recv_sync |
| 36 | Slow_Rate_Group.Tick_ T_Send [8] | Adc_Data_Collector_ Instance.Tick_T_Recv_ Sync | send-recv_sync |
| 37 | Fast_Rate_Group.Tick_ T_Send [1] | Oscillator_A.Tick_T_ Recv_Sync | send-recv_sync |
| 38 | Fast_Rate_Group.Tick_ T_Send [2] | Oscillator_B.Tick_T_ Recv_Sync | send-recv_sync |
| 39 | Fast_Rate_Group.Tick_ T_Send [3] | Product_Packetizer_ Instance.Tick_T_Recv_ Sync | send-recv_sync |

| | | | |
|---|---|---|---|
| 40 | Watchdog_Rate_Group.<br>Tick_T_Send | Task_Watchdog_<br>Instance.Tick_T_Recv_<br>Sync | send-recv_sync |
| 41 | Tick_Divider_<br>Instance.Event_T_Send | Event_Filter_<br>Instance.Event_T_<br>Recv_Sync | send-recv_sync |
| 42 | Slow_Rate_Group.<br>Event_T_Send | Event_Filter_<br>Instance.Event_T_<br>Recv_Sync | send-recv_sync |
| 43 | Fast_Rate_Group.<br>Event_T_Send | Event_Filter_<br>Instance.Event_T_<br>Recv_Sync | send-recv_sync |
| 44 | Watchdog_Rate_Group.<br>Event_T_Send | Event_Filter_<br>Instance.Event_T_<br>Recv_Sync | send-recv_sync |
| 45 | Command_Router_<br>Instance.Event_T_Send | Event_Filter_<br>Instance.Event_T_<br>Recv_Sync | send-recv_sync |
| 46 | Counter_Instance.<br>Event_T_Send | Event_Filter_<br>Instance.Event_T_<br>Recv_Sync | send-recv_sync |
| 47 | Oscillator_A.Event_T_<br>Send | Event_Filter_<br>Instance.Event_T_<br>Recv_Sync | send-recv_sync |
| 48 | Oscillator_B.Event_T_<br>Send | Event_Filter_<br>Instance.Event_T_<br>Recv_Sync | send-recv_sync |
| 49 | Product_Packetizer_<br>Instance.Event_T_Send | Event_Filter_<br>Instance.Event_T_<br>Recv_Sync | send-recv_sync |
| 50 | Stack_Monitor_<br>Instance.Event_T_Send | Event_Filter_<br>Instance.Event_T_<br>Recv_Sync | send-recv_sync |
| 51 | Queue_Monitor_<br>Instance.Event_T_Send | Event_Filter_<br>Instance.Event_T_<br>Recv_Sync | send-recv_sync |
| 52 | Cpu_Monitor_Instance.<br>Event_T_Send | Event_Filter_<br>Instance.Event_T_<br>Recv_Sync | send-recv_sync |

| | | | |
|---|---|---|---|
| 53 | Event_Filter_ Instance.Event_T_Send | Event_Filter_ Instance.Event_T_ Recv_Sync | send-recv_sync |
| 54 | Event_Limiter_ Instance.Event_T_Send | Event_Filter_ Instance.Event_T_ Recv_Sync | send-recv_sync |
| 55 | Zero_Divider_ Instance.Event_T_Send | Event_Filter_ Instance.Event_T_ Recv_Sync | send-recv_sync |
| 56 | Task_Watchdog_ Instance.Event_T_Send | Event_Filter_ Instance.Event_T_ Recv_Sync | send-recv_sync |
| 57 | Fault_Correction_ Instance.Event_T_Send | Event_Filter_ Instance.Event_T_ Recv_Sync | send-recv_sync |
| 58 | Ccsds_Serial_ Interface_Instance. Event_T_Send | Event_Filter_ Instance.Event_T_ Recv_Sync | send-recv_sync |
| 59 | Event_Filter_ Instance.Event_ Forward_T_Send | Event_Limiter_ Instance.Event_T_ Recv_Sync | send-recv_sync |
| 60 | Event_Limiter_ Instance.Event_ Forward_T_Send | Event_Packetizer_ Instance.Event_T_ Recv_Sync | send-recv_sync |
| 61 | Product_Database_ Instance.Event_T_Send | Event_Filter_ Instance.Event_T_ Recv_Sync | send-recv_sync |
| 62 | Ccsds_Command_ Depacketizer_ Instance.Event_T_Send | Event_Filter_ Instance.Event_T_ Recv_Sync | send-recv_sync |
| 63 | Fault_Producer_ Instance.Event_T_Send | Event_Filter_ Instance.Event_T_ Recv_Sync | send-recv_sync |
| 64 | Command_Router_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 65 | Counter_Instance. Command_Response_T_ Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |

| | | | |
|---|---|---|---|
| 66 | Oscillator_A.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 67 | Oscillator_B.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 68 | Product_Packetizer_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 69 | Event_Packetizer_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 70 | Ccsds_Command_ Depacketizer_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 71 | Product_Database_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 72 | Stack_Monitor_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 73 | Queue_Monitor_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 74 | Cpu_Monitor_Instance. Command_Response_T_ Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 75 | Event_Filter_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 76 | Event_Limiter_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 77 | Task_Watchdog_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 78 | Fault_Correction_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |

| | | | |
|---|---|---|---|
| 79 | Zero_Divider_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 80 | Fault_Producer_ Instance.Command_ Response_T_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 81 | Ccsds_Command_ Depacketizer_ Instance.Command_ T_Send | Command_Router_ Instance.Command_ T_To_Route_Recv_Async | send-recv_async |
| 82 | Command_Router_ Instance.Command_ T_Send [1] | Command_Router_ Instance.Command_ T_Recv_Async | send-recv_async |
| 83 | Command_Router_ Instance.Command_ T_Send [2] | Counter_Instance. Command_T_Recv_Async | send-recv_async |
| 84 | Command_Router_ Instance.Command_ T_Send [3] | Oscillator_A.Command_ T_Recv_Async | send-recv_async |
| 85 | Command_Router_ Instance.Command_ T_Send [4] | Oscillator_B.Command_ T_Recv_Async | send-recv_async |
| 86 | Command_Router_ Instance.Command_ T_Send [5] | Product_Packetizer_ Instance.Command_T_ Recv_Async | send-recv_async |
| 87 | Command_Router_ Instance.Command_ T_Send [6] | Event_Packetizer_ Instance.Command_T_ Recv_Sync | send-recv_sync |
| 88 | Command_Router_ Instance.Command_ T_Send [7] | Product_Database_ Instance.Command_T_ Recv_Sync | send-recv_sync |
| 89 | Command_Router_ Instance.Command_ T_Send [8] | Ccsds_Command_ Depacketizer_ Instance.Command_ T_Recv_Sync | send-recv_sync |
| 90 | Command_Router_ Instance.Command_ T_Send [9] | Stack_Monitor_ Instance.Command_ T_Recv_Sync | send-recv_sync |

| | | | |
|---|---|---|---|
| 91 | Command_Router_ Instance.Command_ T_Send [10] | Queue_Monitor_ Instance.Command_ T_Recv_Sync | send-recv_sync |
| 92 | Command_Router_ Instance.Command_ T_Send [11] | Cpu_Monitor_Instance. Command_T_Recv_Sync | send-recv_sync |
| 93 | Command_Router_ Instance.Command_ T_Send [12] | Event_Filter_ Instance.Command_ T_Recv_Sync | send-recv_sync |
| 94 | Command_Router_ Instance.Command_ T_Send [13] | Event_Limiter_ Instance.Command_ T_Recv_Sync | send-recv_sync |
| 95 | Command_Router_ Instance.Command_ T_Send [14] | Zero_Divider_ Instance.Command_ T_Recv_Sync | send-recv_sync |
| 96 | Command_Router_ Instance.Command_ T_Send [15] | Fault_Correction_ Instance.Command_T_ Recv_Async | send-recv_async |
| 97 | Command_Router_ Instance.Command_ T_Send [16] | Task_Watchdog_ Instance.Command_ T_Recv_Sync | send-recv_sync |
| 98 | Command_Router_ Instance.Command_ T_Send [17] | Fault_Producer_ Instance.Command_ T_Recv_Sync | send-recv_sync |
| 99 | Command_Router_ Instance.Command_ Response_T_To_ Forward_Send | Command_Router_ Instance.Command_ Response_T_Recv_Async | send-recv_async |
| 100 | Oscillator_A.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 101 | Oscillator_B.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 102 | Event_Packetizer_ Instance.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |

| | | | |
|---|---|---|---|
| 103 | Ccsds_Command_ Depacketizer_ Instance.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 104 | Slow_Rate_Group.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 105 | Fast_Rate_Group.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 106 | Watchdog_Rate_Group. Data_Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 107 | Command_Router_ Instance.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 108 | Product_Database_ Instance.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 109 | Stack_Monitor_ Instance.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 110 | Queue_Monitor_ Instance.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 111 | Cpu_Monitor_Instance. Data_Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 112 | Event_Filter_ Instance.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 113 | Event_Limiter_ Instance.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 114 | Task_Watchdog_ Instance.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 115 | Fault_Correction_ Instance.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |

| | | | |
|---|---|---|---|
| 116 | Adc_Data_Collector_ Instance.Data_ Product_T_Send | Product_Database_ Instance.Data_ Product_T_Recv_Sync | send-recv_sync |
| 117 | Product_Packetizer_ Instance.Data_ Product_Fetch_T_ Request | Product_Database_ Instance.Data_ Product_Fetch_T_ Service | request-service |
| 118 | Event_Packetizer_ Instance.Packet_T_ Send | Ccsds_Packetizer_ Instance.Packet_T_ Recv_Sync | send-recv_sync |
| 119 | Product_Packetizer_ Instance.Packet_T_ Send | Ccsds_Packetizer_ Instance.Packet_T_ Recv_Sync | send-recv_sync |
| 120 | Counter_Instance. Packet_T_Send | Ccsds_Packetizer_ Instance.Packet_T_ Recv_Sync | send-recv_sync |
| 121 | Ccsds_Command_ Depacketizer_ Instance.Packet_T_ Send | Ccsds_Packetizer_ Instance.Packet_T_ Recv_Sync | send-recv_sync |
| 122 | Product_Database_ Instance.Packet_T_ Send | Ccsds_Packetizer_ Instance.Packet_T_ Recv_Sync | send-recv_sync |
| 123 | Cpu_Monitor_Instance. Packet_T_Send | Ccsds_Packetizer_ Instance.Packet_T_ Recv_Sync | send-recv_sync |
| 124 | Queue_Monitor_ Instance.Packet_T_ Send | Ccsds_Packetizer_ Instance.Packet_T_ Recv_Sync | send-recv_sync |
| 125 | Stack_Monitor_ Instance.Packet_T_ Send | Ccsds_Packetizer_ Instance.Packet_T_ Recv_Sync | send-recv_sync |
| 126 | Event_Filter_ Instance.Packet_T_ Send | Ccsds_Packetizer_ Instance.Packet_T_ Recv_Sync | send-recv_sync |
| 127 | Event_Limiter_ Instance.Packet_T_ Send | Ccsds_Packetizer_ Instance.Packet_T_ Recv_Sync | send-recv_sync |

| | | | |
|---|---|---|---|
| 128 | Slow_Rate_Group.Pet_T_Send | Task_Watchdog_Instance.Pet_T_Recv_Sync | send-recv_sync |
| 129 | Fast_Rate_Group.Pet_T_Send | Task_Watchdog_Instance.Pet_T_Recv_Sync | send-recv_sync |
| 130 | Task_Watchdog_Instance.Fault_T_Send | Fault_Correction_Instance.Fault_T_Recv_Async | send-recv_async |
| 131 | Fault_Producer_Instance.Fault_T_Send | Fault_Correction_Instance.Fault_T_Recv_Async | send-recv_async |
| 132 | Fault_Correction_Instance.Command_T_Send | Command_Router_Instance.Command_T_To_Route_Recv_Sync | send-recv_sync |
| 133 | Ccsds_Packetizer_Instance.Ccsds_Space_Packet_T_Send | Ccsds_Serial_Interface_Instance.Ccsds_Space_Packet_T_Recv_Async | send-recv_async |
| 134 | Ccsds_Serial_Interface_Instance.Ccsds_Space_Packet_T_Send | Ccsds_Command_Depacketizer_Instance.Ccsds_Space_Packet_T_Recv_Sync | send-recv_sync |

Connection Descriptions:

- **Ticker_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Tick_Divider_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Slow_Rate_Group.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Fast_Rate_Group.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Watchdog_Rate_Group.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Command_Router_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Counter_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Oscillator_A.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Oscillator_B.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*

- **Event_Packetizer_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Product_Database_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Product_Packetizer_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Ccsds_Command_Depacketizer_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Cpu_Monitor_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Stack_Monitor_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Queue_Monitor_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Event_Filter_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Event_Limiter_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Zero_Divider_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Task_Watchdog_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Fault_Correction_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Fault_Producer_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Ccsds_Serial_Interface_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Adc_Data_Collector_Instance.Sys_Time_T_Get-System_Time_Instance.Sys_Time_T_Return** - *No description provided.*
- **Ticker_Instance.Tick_T_Send-Tick_Divider_Instance.Tick_T_Recv_Sync** - *No description provided.*
- **Tick_Divider_Instance.Tick_T_Send[1]-Watchdog_Rate_Group.Tick_T_Recv_Async** - *No description provided.*
- **Tick_Divider_Instance.Tick_T_Send[2]-Slow_Rate_Group.Tick_T_Recv_Async** - *No description provided.*
- **Tick_Divider_Instance.Tick_T_Send[3]-Fast_Rate_Group.Tick_T_Recv_Async** - *No description provided.*
- **Slow_Rate_Group.Tick_T_Send[1]-Counter_Instance.Tick_T_Recv_Sync** - Schedule connection from the rate group to the counter
- **Slow_Rate_Group.Tick_T_Send[2]-Event_Packetizer_Instance.Tick_T_Recv_Sync** - Schedule connection from the rate group to the packetizer
- **Slow_Rate_Group.Tick_T_Send[3]-Cpu_Monitor_Instance.Tick_T_Recv_Sync** - *No description provided.*
- **Slow_Rate_Group.Tick_T_Send[4]-Queue_Monitor_Instance.Tick_T_Recv_Sync** - *No description provided.*
- **Slow_Rate_Group.Tick_T_Send[5]-Stack_Monitor_Instance.Tick_T_Recv_**

**Sync** - *No description provided.*

- **Slow_Rate_Group.Tick_T_Send[6]-Event_Filter_Instance.Tick_T_Recv_Sync** - *No description provided.*
- **Slow_Rate_Group.Tick_T_Send[7]-Event_Limiter_Instance.Tick_T_Recv_Sync** - *No description provided.*
- **Slow_Rate_Group.Tick_T_Send[8]-Adc_Data_Collector_Instance.Tick_T_Recv_Sync** - *No description provided.*
- **Fast_Rate_Group.Tick_T_Send[1]-Oscillator_A.Tick_T_Recv_Sync** - Schedule connection from the rate group to the oscillator
- **Fast_Rate_Group.Tick_T_Send[2]-Oscillator_B.Tick_T_Recv_Sync** - Schedule connection from the rate group to the oscillator
- **Fast_Rate_Group.Tick_T_Send[3]-Product_Packetizer_Instance.Tick_T_Recv_Sync** - Schedule connection from the rate group to the packetizer
- **Watchdog_Rate_Group.Tick_T_Send-Task_Watchdog_Instance.Tick_T_Recv_Sync** - *No description provided.*
- **Tick_Divider_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Slow_Rate_Group.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Fast_Rate_Group.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Watchdog_Rate_Group.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Counter_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Oscillator_A.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Oscillator_B.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Product_Packetizer_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Stack_Monitor_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Queue_Monitor_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Cpu_Monitor_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Event_Filter_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Event_Limiter_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Zero_Divider_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Task_Watchdog_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*

- **Fault_Correction_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Ccsds_Serial_Interface_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Event_Filter_Instance.Event_Forward_T_Send-Event_Limiter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Event_Limiter_Instance.Event_Forward_T_Send-Event_Packetizer_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Product_Database_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Ccsds_Command_Depacketizer_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Fault_Producer_Instance.Event_T_Send-Event_Filter_Instance.Event_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Counter_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Oscillator_A.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Oscillator_B.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Product_Packetizer_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Event_Packetizer_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Ccsds_Command_Depacketizer_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Product_Database_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Stack_Monitor_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Queue_Monitor_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Cpu_Monitor_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Event_Filter_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Event_Limiter_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Task_Watchdog_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Fault_Correction_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Zero_Divider_Instance.Command_Response_T_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Fault_Producer_Instance.Command_Response_T_Send-Command_Router_**

**Instance.Command_Response_T_Recv_Async** - *No description provided.*

- **Ccsds_Command_Depacketizer_Instance.Command_T_Send-Command_Router_Instance.Command_T_To_Route_Recv_Async** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[1]-Command_Router_Instance.Command_T_Recv_Async** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[2]-Counter_Instance.Command_T_Recv_Async** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[3]-Oscillator_A.Command_T_Recv_Async** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[4]-Oscillator_B.Command_T_Recv_Async** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[5]-Product_Packetizer_Instance.Command_T_Recv_Async** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[6]-Event_Packetizer_Instance.Command_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[7]-Product_Database_Instance.Command_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[8]-Ccsds_Command_Depacketizer_Instance.Command_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[9]-Stack_Monitor_Instance.Command_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[10]-Queue_Monitor_Instance.Command_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[11]-Cpu_Monitor_Instance.Command_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[12]-Event_Filter_Instance.Command_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[13]-Event_Limiter_Instance.Command_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[14]-Zero_Divider_Instance.Command_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[15]-Fault_Correction_Instance.Command_T_Recv_Async** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[16]-Task_Watchdog_Instance.Command_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Command_T_Send[17]-Fault_Producer_Instance.Command_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Command_Response_T_To_Forward_Send-Command_Router_Instance.Command_Response_T_Recv_Async** - *No description provided.*
- **Oscillator_A.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - Data product connection between the oscillator and packetizer
- **Oscillator_B.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - Data product connection between the oscillator and packetizer
- **Event_Packetizer_Instance.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Ccsds_Command_Depacketizer_Instance.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*

- **Slow_Rate_Group.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Fast_Rate_Group.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Watchdog_Rate_Group.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Command_Router_Instance.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Product_Database_Instance.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Stack_Monitor_Instance.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Queue_Monitor_Instance.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Cpu_Monitor_Instance.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Event_Filter_Instance.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Event_Limiter_Instance.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Task_Watchdog_Instance.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Fault_Correction_Instance.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Adc_Data_Collector_Instance.Data_Product_T_Send-Product_Database_Instance.Data_Product_T_Recv_Sync** - *No description provided.*
- **Product_Packetizer_Instance.Data_Product_Fetch_T_Request-Product_Database_Instance.Data_Product_Fetch_T_Service** - *No description provided.*
- **Event_Packetizer_Instance.Packet_T_Send-Ccsds_Packetizer_Instance.Packet_T_Recv_Sync** - *No description provided.*
- **Product_Packetizer_Instance.Packet_T_Send-Ccsds_Packetizer_Instance.Packet_T_Recv_Sync** - *No description provided.*
- **Counter_Instance.Packet_T_Send-Ccsds_Packetizer_Instance.Packet_T_Recv_Sync** - *No description provided.*
- **Ccsds_Command_Depacketizer_Instance.Packet_T_Send-Ccsds_Packetizer_Instance.Packet_T_Recv_Sync** - *No description provided.*
- **Product_Database_Instance.Packet_T_Send-Ccsds_Packetizer_Instance.Packet_T_Recv_Sync** - *No description provided.*
- **Cpu_Monitor_Instance.Packet_T_Send-Ccsds_Packetizer_Instance.Packet_T_Recv_Sync** - *No description provided.*
- **Queue_Monitor_Instance.Packet_T_Send-Ccsds_Packetizer_Instance.Packet_T_Recv_Sync** - *No description provided.*
- **Stack_Monitor_Instance.Packet_T_Send-Ccsds_Packetizer_Instance.Packet_T_Recv_Sync** - *No description provided.*
- **Event_Filter_Instance.Packet_T_Send-Ccsds_Packetizer_Instance.Packet_T_Recv_Sync** - *No description provided.*
- **Event_Limiter_Instance.Packet_T_Send-Ccsds_Packetizer_Instance.**

**Packet_T_Recv_Sync** - *No description provided.*

- **Slow_Rate_Group.Pet_T_Send-Task_Watchdog_Instance.Pet_T_Recv_Sync[1]** - *No description provided.*
- **Fast_Rate_Group.Pet_T_Send-Task_Watchdog_Instance.Pet_T_Recv_Sync[2]** - *No description provided.*
- **Task_Watchdog_Instance.Fault_T_Send-Fault_Correction_Instance.Fault_T_Recv_Async** - *No description provided.*
- **Fault_Producer_Instance.Fault_T_Send-Fault_Correction_Instance.Fault_T_Recv_Async** - *No description provided.*
- **Fault_Correction_Instance.Command_T_Send-Command_Router_Instance.Command_T_To_Route_Recv_Sync** - We connect the fault correction command response to the command router synchronous connector for the fastest, most reliable execution. This bypasses the command router's queue.
- **Ccsds_Packetizer_Instance.Ccsds_Space_Packet_T_Send-Ccsds_Serial_Interface_Instance.Ccsds_Space_Packet_T_Recv_Async** - *No description provided.*
- **Ccsds_Serial_Interface_Instance.Ccsds_Space_Packet_T_Send-Ccsds_Command_Depacketizer_Instance.Ccsds_Space_Packet_T_Recv_Sync** - *No description provided.*

## 3.2 Packed Types

The following section outlines any complex data types used in the assembly in alphabetical order. This includes packed records and packed arrays that might be used as connector types, command arguments, or event parameters.

### Ccsds_Command_Header.T:

Record for a LASP-specific CCSDS command header.

Table 11: Ccsds_Command_Header Packed Record : 64 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Primary_Header | Ccsds_Primary_Header.T | - | 48 | 0 | 47 |
| Secondary_Header | Ccsds_Command_Secondary_Header.T | - | 16 | 48 | 63 |

Field Descriptions:

- **Primary_Header** - The CCSDS primary header
- **Secondary_Header** - The command secondary header

### Ccsds_Command_Secondary_Header.T:

Record for the LASP-specific command secondary header.

*Preamble (inline Ada definitions):*

```
type Function_Code_Type is mod 2**7;
type One_Bit_Pad_Type is mod 2**1;
```

Table 12: Ccsds_Command_Secondary_Header Packed Record : 16 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Reserved | One_Bit_Pad_Type | 0 to 1 | 1 | 0 | 0 |
| Function_Code | Function_Code_Type | 0 to 127 | 7 | 1 | 7 |
| Checksum | Interfaces.Unsigned_8 | 0 to 255 | 8 | 8 | 15 |

Field Descriptions:

- **Reserved** - Reserve bit.
- **Function_Code** - The command function code.
- **Checksum** - An 8 bit checksum over the entire command packet

## Ccsds_Primary_Header.T:

Record for the CCSDS Packet Primary Header

*Preamble (inline Ada definitions):*

```
subtype Three_Bit_Version_Type is Interfaces.Unsigned_8 range 0 .. 7;
type Ccsds_Apid_Type is mod 2**11;
type Ccsds_Sequence_Count_Type is mod 2**14;
```

Table 13: Ccsds_Primary_Header Packed Record : 48 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Version | Three_Bit_Version_Type | 0 to 7 | 3 | 0 | 2 |
| Packet_Type | Ccsds_Enums.Ccsds_Packet_Type.E | 0 => Telemetry<br>1 => Telecommand | 1 | 3 | 3 |
| Secondary_Header | Ccsds_Enums.Ccsds_Secondary_Header_Indicator.E | 0 => Secondary_Header_Not_Present<br>1 => Secondary_Header_Present | 1 | 4 | 4 |
| Apid | Ccsds_Apid_Type | 0 to 2047 | 11 | 5 | 15 |
| Sequence_Flag | Ccsds_Enums.Ccsds_Sequence_Flag.E | 0 => Continuationsegment<br>1 => Firstsegment<br>2 => Lastsegment<br>3 => Unsegmented | 2 | 16 | 17 |

| | | | | | |
|---|---|---|---|---|---|
| Sequence_ Count | Ccsds_ Sequence_ Count_ Type | 0 to 16383 | 14 | 18 | 31 |
| Packet_ Length | Interfaces. Unsigned_ 16 | 0 to 65535 | 16 | 32 | 47 |

Field Descriptions:

- **Version** - Packet Version Number
- **Packet_Type** - Packet Type
- **Secondary_Header** - Does packet have CCSDS secondary header
- **Apid** - Application process identifier
- **Sequence_Flag** - Sequence Flag
- **Sequence_Count** - Packet Sequence Count
- **Packet_Length** - This is the packet data length. One added to this number corresponds to the number of bytes included in the data section of the CCSDS Space Packet.

## Ccsds_Space_Packet.T:

Record for the CCSDS Space Packet

*Preamble (inline Ada definitions):*

```
use Basic_Types;
subtype Ccsds_Data_Type is Byte_Array (0 ..
 ↪  Configuration.Ccsds_Packet_Buffer_Size - 1);
```

Table 14: Ccsds_Space_Packet Packed Record : 10240 bits *(maximum)*

| Name | Type | Range | Size (Bits) | Start Bit | End Bit | Variable Length |
|---|---|---|---|---|---|---|
| Header | Ccsds_ Primary_ Header.T | - | 48 | 0 | 47 | - |
| Data | Ccsds_Data_ Type | - | 10192 | 48 | 10239 | Header. Packet_Length |

Field Descriptions:

- **Header** - The CCSDS Primary Header
- **Data** - User Data Field

## Command.T:

Generic command packet for holding arbitrary commands

Table 15: Command Packed Record : 2080 bits *(maximum)*

| Name | Type | Range | Size (Bits) | Start Bit | End Bit | Variable Length |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| Header | Command_ Header.T | - | 40 | 0 | 39 | - |
| Arg_Buffer | Command_ Types. Command_Arg_ Buffer_Type | - | 2040 | 40 | 2079 | Header.Arg_ Buffer_Length |

Field Descriptions:

- **Header** - The command header
- **Arg_Buffer** - A buffer to that contains the command arguments

## Command_Header.T:

Generic command header for holding arbitrary commands

Table 16: Command_Header Packed Record : 40 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Source_Id | Command_Types. Command_Source_Id | 0 to 65535 | 16 | 0 | 15 |
| Id | Command_Types. Command_Id | 0 to 65535 | 16 | 16 | 31 |
| Arg_Buffer_Length | Command_Types. Command_Arg_Buffer_ Length_Type | 0 to 255 | 8 | 32 | 39 |

Field Descriptions:

- **Source_Id** - The source ID. An ID assigned to a command sending component.
- **Id** - The command identifier
- **Arg_Buffer_Length** - The number of bytes used in the command argument buffer

## Command_Id.T:

A packed record which holds a command identifier.

Table 17: Command_Id Packed Record : 16 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Command_Types. Command_Id | 0 to 65535 | 16 | 0 | 15 |

Field Descriptions:

- **Id** - The command identifier

## Command_Id_Status.T:

Record for holding a command identifier and command response status.

Table 18: Command_Id_Status Packed Record : 24 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Command_Types.<br>Command_Id | 0 to 65535 | 16 | 0 | 15 |
| Status | Command_Enums.<br>Command_<br>Response_<br>Status.E | 0 => Success<br>1 => Failure<br>2 => Id_Error<br>3 => Validation_Error<br>4 => Length_Error<br>5 => Dropped<br>6 => Register<br>7 => Register_Source | 8 | 16 | 23 |

Field Descriptions:

- **Id** - The command ID for the command response.
- **Status** - The command execution status.

## Command_Response.T:

Record for holding command response data.

Table 19: Command_Response Packed Record : 56 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Source_Id | Command_<br>Types.Command_<br>Source_Id | 0 to 65535 | 16 | 0 | 15 |
| Registration_<br>Id | Command_<br>Types.Command_<br>Registration_<br>Id | 0 to 65535 | 16 | 16 | 31 |
| Command_Id | Command_Types.<br>Command_Id | 0 to 65535 | 16 | 32 | 47 |
| Status | Command_Enums.<br>Command_<br>Response_<br>Status.E | 0 => Success<br>1 => Failure<br>2 => Id_Error<br>3 => Validation_Error<br>4 => Length_Error<br>5 => Dropped<br>6 => Register<br>7 => Register_Source | 8 | 48 | 55 |

Field Descriptions:

- **Source_Id** - The source ID. An ID assigned to a command sending component.
- **Registration_Id** - The registration ID. An ID assigned to each registered component at initialization.
- **Command_Id** - The command ID for the command response.
- **Status** - The command execution status.

## Command_Router_Arg.T:

A 32-bit unsigned integer with range 0 to 999.

*Preamble (inline Ada definitions):*

```
subtype Value_Type is Natural range 0 .. 999;
```

Table 20: Command_Router_Arg Packed Record : 32 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Value | Value_Type | 0 to 999 | 32 | 0 | 31 |

Field Descriptions:

- **Value** - The 32-bit unsigned integer with range 0 to 999.

## Cycle_Slip_Param.T:

This is a type that contains useful information about a cycle slip.

Table 21: Cycle_Slip_Param Packed Record : 112 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Slipped_Tick | Tick.T | - | 96 | 0 | 95 |
| Num_Slips | Interfaces. Unsigned_16 | 0 to 65535 | 16 | 96 | 111 |

Field Descriptions:

- **Slipped_Tick** - The tick during which the cycle slip occured.
- **Num_Slips** - The number of cycle slips that have occured.

## Data_Product.T:

Generic data product packet for holding arbitrary data types

Table 22: Data_Product Packed Record : 344 bits *(maximum)*

| Name | Type | Range | Size (Bits) | Start Bit | End Bit | Variable Length |
|---|---|---|---|---|---|---|
| Header | Data_Product_ Header.T | - | 88 | 0 | 87 | - |
| Buffer | Data_Product_ Types.Data_ Product_ Buffer_Type | - | 256 | 88 | 343 | Header.Buffer_ Length |

Field Descriptions:

- **Header** - The data product header
- **Buffer** - A buffer that contains the data product type

## Data_Product_Fetch.T:

A packed record which holds information for a data product request.

Table 23: Data_Product_Fetch Packed Record : 16 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Data_Product_Types.Data_Product_Id | 0 to 65535 | 16 | 0 | 15 |

Field Descriptions:
- **Id** - The data product identifier

## Data_Product_Header.T:

Generic data_product packet for holding arbitrary data_product types

Table 24: Data_Product_Header Packed Record : 88 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Time | Sys_Time.T | - | 64 | 0 | 63 |
| Id | Data_Product_Types.Data_Product_Id | 0 to 65535 | 16 | 64 | 79 |
| Buffer_Length | Data_Product_Types.Data_Product_Buffer_Length_Type | 0 to 32 | 8 | 80 | 87 |

Field Descriptions:
- **Time** - The timestamp for the data product item.
- **Id** - The data product identifier
- **Buffer_Length** - The number of bytes used in the data product buffer

## Data_Product_Id.T:

A packed record which holds a data product identifier.

Table 25: Data_Product_Id Packed Record : 16 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Data_Product_Types.Data_Product_Id | 0 to 65535 | 16 | 0 | 15 |

Field Descriptions:
- **Id** - The data product identifier

## Data_Product_Poly_Event.T:

Data product with 4 byte data buffer.

Table 26: Data_Product_Poly_Event Packed Record : 120 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Header | Data_Product_Header.T | - | 88 | 0 | 87 |
| Data | Basic_Types.Poly_32_Type | - | 32 | 88 | 119 |

Field Descriptions:

- **`Header`** - The data product header
- **`Data`** - The polymorphic type.

## Data_Product_Poly_Extract.T:

Contains information to extract a poly type from a data product.

*Preamble (inline Ada definitions):*

```
subtype Data_Product_Bit_Offset_Type is Natural range Natural'First ..
↪  Data_Product_Types.Data_Product_Buffer_Type'Length *
↪  Basic_Types.Byte'Object_Size; subtype Poly_Type_Size_Type is Positive range
↪  Positive'First .. 32;
```

Table 27: Data_Product_Poly_Extract Packed Record : 40 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Data_Product_Types.Data_Product_Id | 0 to 65535 | 16 | 0 | 15 |
| Offset | Data_Product_Bit_Offset_Type | 0 to 256 | 16 | 16 | 31 |
| Size | Poly_Type_Size_Type | 1 to 32 | 8 | 32 | 39 |

Field Descriptions:

- **`Id`** - ID of the data product.
- **`Offset`** - Offset of the data product item (in bits).
- **`Size`** - Size of the data product item (in bits).

## Data_Product_Poly_Type.T:

Data product poly type, for dumping arbitrary data products.

Table 28: Data_Product_Poly_Type Packed Record : 112 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Time | Sys_Time.T | - | 64 | 0 | 63 |
| Id | Data_Product_Types.Data_Product_Id | 0 to 65535 | 16 | 64 | 79 |
| Data | Basic_Types.Poly_32_Type | - | 32 | 80 | 111 |

Field Descriptions:

- **`Time`** - The timestamp for the data product item.
- **`Id`** - The data product identifier
- **`Data`** - The polymorphic type.

## Data_Product_Return.T:

This record holds data returned from a data product fetch request.

Table 29: Data_Product_Return Packed Record : 352 bits *(maximum)*

| Name | Type | Range | Size (Bits) | Start Bit | End Bit | Variable Length |
|---|---|---|---|---|---|---|
| The_Status | Data_Product_Enums.Fetch_Status.E | 0 => Success<br>1 => Not_Available<br>2 => Id_Out_Of_Range | 8 | 0 | 7 | - |
| The_Data_Product | Data_Product.T | - | 344 | 8 | 351 | - |

Field Descriptions:

- **The_Status** - A status relating whether or not the data product fetch was successful or not.
- **The_Data_Product** - The data product item returned.

## Delta_Time.T:

A record which holds a time difference using GPS format including seconds and subseconds.

Table 30: Delta_Time Packed Record : 64 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Seconds | Interfaces.Unsigned_32 | 0 to 4294967295 | 32 | 0 | 31 |
| Subseconds | Interfaces.Unsigned_32 | 0 to 4294967295 | 32 | 32 | 63 |

Field Descriptions:

- **Seconds** - The number of seconds elapsed since epoch.
- **Subseconds** - The number of 1/(2^32) sub-seconds.

## Event.T:

Generic event packet for holding arbitrary events

Table 31: Event Packed Record : 344 bits *(maximum)*

| Name | Type | Range | Size (Bits) | Start Bit | End Bit | Variable Length |
|---|---|---|---|---|---|---|
| Header | Event_Header.T | - | 88 | 0 | 87 | - |
| Param_Buffer | Event_Types.Parameter_Buffer_Type | - | 256 | 88 | 343 | Header.Param_Buffer_Length |

Field Descriptions:

- **Header** - The event header
- **Param_Buffer** - A buffer that contains the event parameters

## Event_Component_State_Type.T:

This record is for the data product that indicates if the event filter component is enabled for filtering or disabled all together.

Table 32: Event_Component_State_Type Packed Record : 8 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Component_Filter_State | Event_Filter_Entry_Enums.Global_Filter_State.E | 0 => Disabled<br>1 => Enabled | 8 | 0 | 7 |

Field Descriptions:

- **Component_Filter_State** - Flag to indicate if the component is enabled or disabled at a overriding level from the individal event states

## Event_Enable_State_Type.T:

This record contains the definition for a two event ID type for ranges in the event limiter commands as well as an issue packet type for issuing packets

Table 33: Event_Enable_State_Type Packed Record : 8 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Component_Enable_State | Two_Counter_Entry_Enums.Event_State_Type.E | 0 => Disabled<br>1 => Enabled | 8 | 0 | 7 |

Field Descriptions:

- **Component_Enable_State** - Flag to indicate if the component is enabled or disabled at a overriding level

## Event_Filter_Id_Range.T:

This record contains the definition for a two event ID type for ranges in the event limiter commands as well as an issue packet type for issuing packets

Table 34: Event_Filter_Id_Range Packed Record : 40 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Start_Event_Id | Event_Id.T | - | 16 | 0 | 15 |
| Stop_Event_Id | Event_Id.T | - | 16 | 16 | 31 |
| Issue_State_Packet | Event_Filter_Enums.Issue_Packet_Type.E | 0 => No_Issue<br>1 => Issue | 8 | 32 | 39 |

Field Descriptions:

- **Start_Event_Id** - The starting event ID to begin the range
- **Stop_Event_Id** - The last event ID to end the range

- **Issue_State_Packet** - Flag to indicate if we dump the states after the change is complete

## Event_Filter_Single_Event_Cmd_Type.T:

This record contains the definition for a two event ID type for ranges in the event limiter commands as well as an issue packet type for issuing packets

Table 35: Event_Filter_Single_Event_Cmd_Type Packed Record : 24 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Event_To_Update | Event_Id.T | - | 16 | 0 | 15 |
| Issue_State_Packet | Event_Filter_ Enums.Issue_ Packet_Type.E | 0 => No_Issue 1 => Issue | 8 | 16 | 23 |

Field Descriptions:

- **Event_To_Update** - The starting event ID to begin the range
- **Issue_State_Packet** - Flag to indicate if we dump the states after the change is complete

## Event_Header.T:

Generic event packet for holding arbitrary events

Table 36: Event_Header Packed Record : 88 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Time | Sys_Time.T | - | 64 | 0 | 63 |
| Id | Event_Types.Event_ Id | 0 to 65535 | 16 | 64 | 79 |
| Param_Buffer_Length | Event_Types. Parameter_Buffer_ Length_Type | 0 to 32 | 8 | 80 | 87 |

Field Descriptions:

- **Time** - The timestamp for the event.
- **Id** - The event identifier
- **Param_Buffer_Length** - The number of bytes used in the param buffer

## Event_Id.T:

A packed record which holds an event identifier.

Table 37: Event_Id Packed Record : 16 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Event_Types.Event_ Id | 0 to 65535 | 16 | 0 | 15 |

Field Descriptions:

- **Id** - The event identifier

## Event_Id_Array.T:

Packed array of events so that it can be used with the event for which ids were limited

Table 38: Event_Id_Array Packed Array : 160 bits

| Type | Range | Element Size (Bits) | Length | Total Size (Bits) |
|---|---|---|---|---|
| **Event_Types.Event_Id** | 0 to 65535 | 16 | 10 | 160 |

## Event_Limiter_Id_Range.T:

This record contains the definition for a two event ID type for ranges in the event limiter commands as well as an issue packet type for issuing packets

Table 39: Event_Limiter_Id_Range Packed Record : 40 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Start_Event_Id | Event_Id.T | - | 16 | 0 | 15 |
| Stop_Event_Id | Event_Id.T | - | 16 | 16 | 31 |
| Issue_State_Packet | Event_Limiter_ Enums.Issue_ Packet_Type.E | 0 => No_Issue<br>1 => Issue | 8 | 32 | 39 |

Field Descriptions:

- **Start_Event_Id** - The starting event ID to begin the range
- **Stop_Event_Id** - The last event ID to end the range
- **Issue_State_Packet** - Flag to indicate if we dump the states after the change is complete

## Event_Limiter_Num_Events_Type.T:

This record contains the definition for the format of the event for this component that contains up to 10 event IDs of those that were limited.

Table 40: Event_Limiter_Num_Events_Type Packed Record : 184 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Num_Events_Limited | Interfaces. Unsigned_16 | 0 to 65535 | 16 | 0 | 15 |
| Num_Event_Ids | Interfaces. Unsigned_8 | 0 to 255 | 8 | 16 | 23 |
| Event_Id_Limited_ Array | Event_Id_Array.T | - | 160 | 24 | 183 |

Field Descriptions:

- **Num_Events_Limited** - The number of events limited since last issuing the event.
- **Num_Event_Ids** - The number of event IDs contained in the event that indicate that event was limited.
- **Event_Id_Limited_Array** - The Array that list the event IDs that were limited since the last event message.

## Event_Limiter_Persistence_Type.T:

This record contains the definition for a packed persistence type

Table 41: Event_Limiter_Persistence_Type Packed Record : 8 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Persistence | Two_Counter_Entry.<br>Persistence_Type | 1 to 7 | 8 | 0 | 7 |

Field Descriptions:

- **Persistence** - The persistence that is used when limiting events.

## Event_Single_State_Cmd_Type.T:

This record contains the definition for a two event ID type for ranges in the event limiter commands as well as an issue packet type for issuing packets

Table 42: Event_Single_State_Cmd_Type Packed Record : 24 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Event_To_Update | Event_Id.T | - | 16 | 0 | 15 |
| Issue_State_Packet | Event_Limiter_<br>Enums.Issue_<br>Packet_Type.E | 0 => No_Issue<br>1 => Issue | 8 | 16 | 23 |

Field Descriptions:

- **Event_To_Update** - The starting event ID to begin the range
- **Issue_State_Packet** - Flag to indicate if we dump the states after the change is complete

## Fault.T:

Generic fault packet for holding arbitrary faults.

Table 43: Fault Packed Record : 152 bits *(maximum)*

| Name | Type | Range | Size (Bits) | Start Bit | End Bit | Variable Length |
|---|---|---|---|---|---|---|
| Header | Fault_Header.T | - | 88 | 0 | 87 | - |
| Param_Buffer | Fault_Types.<br>Parameter_<br>Buffer_Type | - | 64 | 88 | 151 | Header.Param_<br>Buffer_Length |

Field Descriptions:

- **Header** - The fault header
- **Param_Buffer** - A buffer that contains the fault parameters

## Fault_Header.T:

Generic fault header.

Table 44: Fault_Header Packed Record : 88 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Time | Sys_Time.T | - | 64 | 0 | 63 |
| Id | Fault_Types.Fault_Id | 0 to 65535 | 16 | 64 | 79 |
| Param_Buffer_Length | Fault_Types.Parameter_Buffer_Length_Type | 0 to 8 | 8 | 80 | 87 |

Field Descriptions:

- **Time** - The timestamp for the fault.
- **Id** - The fault identifier
- **Param_Buffer_Length** - The number of bytes used in the param buffer

## Fault_Static.T:

Generic fault packet for holding arbitrary faults. This is the same as the Fault.T type, except that it is not variable sized, it is always maximum sized. This can be useful for sending events with faults in them.

Table 45: Fault_Static Packed Record : 152 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Header | Fault_Header.T | - | 88 | 0 | 87 |
| Param_Buffer | Fault_Types.Parameter_Buffer_Type | - | 64 | 88 | 151 |

Field Descriptions:

- **Header** - The fault header
- **Param_Buffer** - A buffer that contains the fault parameters

## Full_Queue_Param.T:

This is a type that contains useful information about a component full queue.

Table 46: Full_Queue_Param Packed Record : 112 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Dropped_Tick | Tick.T | - | 96 | 0 | 95 |
| Index | Connector_Types.Connector_Index_Type | 1 to 65535 | 16 | 96 | 111 |

Field Descriptions:

- **Dropped_Tick** - The tick during which the component's queue was found to be full.
- **Index** - The rate group index number of the component that had a full queue.

## Invalid_Command_Info.T:

Record for holding information about an invalid command

Table 47: Invalid_Command_Info Packed Record : 112 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Command_Types.Command_Id | 0 to 65535 | 16 | 0 | 15 |
| Errant_Field_Number | Interfaces.Unsigned_32 | 0 to 4294967295 | 32 | 16 | 47 |
| Errant_Field | Basic_Types.Poly_Type | - | 64 | 48 | 111 |

Field Descriptions:

- **Id** - The command Id received.
- **Errant_Field_Number** - The field that was invalid. 1 is the first field, 0 means unkown field, 2**32 means that the length field of the command was invalid.
- **Errant_Field** - A polymorphic type containing the bad field data, or length when Errant_Field_Number is 2**32.

## Invalid_Data_Product_Length.T:

A packed record which holds data related to an invalid data product length.

Table 48: Invalid_Data_Product_Length Packed Record : 96 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Header | Data_Product_Header.T | - | 88 | 0 | 87 |
| Expected_Length | Data_Product_Types.Data_Product_Buffer_Length_Type | 0 to 32 | 8 | 88 | 95 |

Field Descriptions:

- **Header** - The packet identifier
- **Expected_Length** - The packet length bound that the length failed to meet.

## Invalid_Packet_Id.T:

A packed record which holds a packet identifier and data product identifier

Table 49: Invalid_Packet_Id Packed Record : 32 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Packet_Id | Packet_Types.Packet_Id | 0 to 65535 | 16 | 0 | 15 |
| Command_Id | Command_Types.Command_Id | 0 to 65535 | 16 | 16 | 31 |

Field Descriptions:

- **Packet_Id** - The packet identifier
- **Command_Id** - The command id

## Invalid_Packet_Length.T:

A packed record which holds data related to an invalid command packet length.

Table 50: Invalid_Packet_Length Packed Record : 112 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Ccsds_Header | Ccsds_ Primary_ Header.T | - | 48 | 0 | 47 |
| Length | Integer | -2147483648 to 2147483647 | 32 | 48 | 79 |
| Length_Bound | Integer | -2147483648 to 2147483647 | 32 | 80 | 111 |

Field Descriptions:

- **Ccsds_Header** - The packet identifier
- **Length** - The packet length
- **Length_Bound** - The packet length bound that the length failed to meet.

## Invalid_Packet_Xor8_Info.T:

A packed record which holds data related to an invalid checksummed CCSDS command packet.

Table 51: Invalid_Packet_Xor8_Info Packed Record : 80 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Ccsds_Header | Ccsds_Command_ Header.T | - | 64 | 0 | 63 |
| Computed_Checksum | Xor_8.Xor_8_Type | 0 to 255 | 8 | 64 | 71 |
| Expected_Checksum | Xor_8.Xor_8_Type | 0 to 255 | 8 | 72 | 79 |

Field Descriptions:

- **Ccsds_Header** - The CCSDS command header.
- **Computed_Checksum** - The computed XOR of the entire packet. This should be 0 if the packet passes.
- **Expected_Checksum** - The XOR included in the CCSDS packet secondary header.

## Invalid_Parameter_Info.T:

Record for holding information about an invalid parameter

Table 52: Invalid_Parameter_Info Packed Record : 112 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Parameter_Types. Parameter_Id | 0 to 65535 | 16 | 0 | 15 |
| Errant_Field_ Number | Interfaces. Unsigned_32 | 0 to 4294967295 | 32 | 16 | 47 |
| Errant_Field | Basic_Types.Poly_ Type | - | 64 | 48 | 111 |

Field Descriptions:

- **Id** - The parameter Id received.

- **Errant_Field_Number** - The field that was invalid. 1 is the first field, 0 means unkown field, 2**32 means that the length field of the parameter was invalid.
- **Errant_Field** - A polymorphic type containing the bad field data, or length when Errant_Field_Number is 2**32.

## Operands.T:

This is a type that contains a left and right side

Table 53: Operands Packed Record : 32 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Left | Interfaces. Unsigned_16 | 0 to 65535 | 16 | 0 | 15 |
| Right | Interfaces. Unsigned_16 | 0 to 65535 | 16 | 16 | 31 |

Field Descriptions:
- **Left** - The left side
- **Right** - The right side

## Packed_Address.T:

A packed system address.

Table 54: Packed_Address Packed Record : 32 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Address | System.Address | - | 32 | 0 | 31 |

Field Descriptions:
- **Address** - The starting address of the memory region.

## Packed_Connector_Index.T:

Single component record for holding packed connector index.

Table 55: Packed_Connector_Index Packed Record : 16 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Index | Connector_Types. Connector_Index_ Type | 1 to 65535 | 16 | 0 | 15 |

Field Descriptions:
- **Index** - The 16-bit connector index.

## Packed_Enable_Disable_Type.T:

Single component record for holding an enable/disable enumeration.

Table 56: Packed_Enable_Disable_Type Packed Record : 8 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| State | Basic_Enums.Enable_Disable_Type.E | 0 => Disabled<br>1 => Enabled | 8 | 0 | 7 |

Field Descriptions:

- **State** - The 8-bit enable disable enumeration.

## Packed_Exception_Occurrence.T:

Packed record which holds information from an Ada Exception Occurrence type. This is the type passed into the Last Chance Handler when running a full runtime.

*Preamble (inline Ada definitions):*

```
type Exception_Name_Buffer is new Basic_Types.Byte_Array (0 .. 99)
  with Size => 100 * 8,
       Object_Size => 100 * 8;
type Exception_Message_Buffer is new Basic_Types.Byte_Array (0 .. 299)
  with Size => 300 * 8,
       Object_Size => 300 * 8;
```

Table 57: Packed_Exception_Occurrence Packed Record : 6432 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Exception_Name | Exception_Name_Buffer | - | 800 | 0 | 799 |
| Exception_Message | Exception_Message_Buffer | - | 2400 | 800 | 3199 |
| Stack_Trace_Depth | Interfaces.Unsigned_32 | 0 to 4294967295 | 32 | 3200 | 3231 |
| Stack_Trace | Stack_Trace_Addresses.T | - | 3200 | 3232 | 6431 |

Field Descriptions:

- **Exception_Name** - The exception name.
- **Exception_Message** - The exception message.
- **Stack_Trace_Depth** - The depth of the reported stack trace.
- **Stack_Trace** - The stack trace addresses.

## Packed_F32.T:

Single component record for holding packed 32-bit floating point number.

Table 58: Packed_F32 Packed Record : 32 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Value | Short_Float | -3.40282e+38 to 3.40282e+38 | 32 | 0 | 31 |

Field Descriptions:

- **`Value`** - The 32-bit floating point number.

## Packed_Fault_Id.T:

A packed record which holds an fault identifier.

Table 59: Packed_Fault_Id Packed Record : 16 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Fault_Types.Fault_Id | 0 to 65535 | 16 | 0 | 15 |

Field Descriptions:
- **`Id`** - The fault identifier

## Packed_Integer.T:

Single component record for holding packed Integer value.

Table 60: Packed_Integer Packed Record : 32 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Value | Integer | -2147483648 to 2147483647 | 32 | 0 | 31 |

Field Descriptions:
- **`Value`** - The 32-bit Signed Integer.

## Packed_Missed_Pet_Limit.T:

Limit value packed for the task watchdog data products

Table 61: Packed_Missed_Pet_Limit Packed Record : 16 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Limit | Task_Watchdog_Types.Missed_Pet_Limit_Type | 1 to 65534 | 16 | 0 | 15 |

Field Descriptions:
- **`Limit`** - The limit value

## Packed_Natural.T:

Single component record for holding packed Natural value.

Table 62: Packed_Natural Packed Record : 32 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Value | Natural | 0 to 2147483647 | 32 | 0 | 31 |

Field Descriptions:

- **Value** - The 32-bit Natural Integer.

## Packed_U16.T:

Single component record for holding packed unsigned 16-bit value.

Table 63: Packed_U16 Packed Record : 16 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Value | Interfaces. Unsigned_16 | 0 to 65535 | 16 | 0 | 15 |

Field Descriptions:
- **Value** - The 16-bit unsigned integer.

## Packed_U32.T:

Single component record for holding packed unsigned 32-bit value.

Table 64: Packed_U32 Packed Record : 32 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Value | Interfaces. Unsigned_32 | 0 to 4294967295 | 32 | 0 | 31 |

Field Descriptions:
- **Value** - The 32-bit unsigned integer.

## Packed_Watchdog_Component_State.T:

State value packed for the task watchdog data products

Table 65: Packed_Watchdog_Component_State Packed Record : 8 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| State | Task_Watchdog_ Enums.Watchdog_ Enabled_State.E | 0 => Disabled 1 => Enabled | 8 | 0 | 7 |

Field Descriptions:
- **State** - The state of the watchdog component to check all upstream petters.

## Packet.T:

Generic packet for holding arbitrary data

Table 66: Packet Packed Record : 10080 bits *(maximum)*

| Name | Type | Range | Size (Bits) | Start Bit | End Bit | Variable Length |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| Header | Packet_ Header.T | - | 112 | 0 | 111 | - |
| Buffer | Packet_ Types.Packet_ Buffer_Type | - | 9968 | 112 | 10079 | Header. Buffer_Length |

Field Descriptions:

- **Header** - The packet header
- **Buffer** - A buffer that contains the packet data

## Packet_Data_Product_Ids.T:

A packed record which holds a packet identifier and data product identifier.

Table 67: Packet_Data_Product_Ids Packed Record : 32 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Packet_Id | Packet_Types. Packet_Id | 0 to 65535 | 16 | 0 | 15 |
| Data_Product_Id | Data_Product_Types. Data_Product_Id | 0 to 65535 | 16 | 16 | 31 |

Field Descriptions:

- **Packet_Id** - The packet identifier
- **Data_Product_Id** - The data product identifier

## Packet_Header.T:

Generic packet header for holding arbitrary data

Table 68: Packet_Header Packed Record : 112 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Time | Sys_Time.T | - | 64 | 0 | 63 |
| Id | Packet_Types. Packet_Id | 0 to 65535 | 16 | 64 | 79 |
| Sequence_Count | Packet_Types. Sequence_Count_Mod_ Type | 0 to 16383 | 16 | 80 | 95 |
| Buffer_Length | Packet_Types. Packet_Buffer_ Length_Type | 0 to 1246 | 16 | 96 | 111 |

Field Descriptions:

- **Time** - The timestamp for the packet item.
- **Id** - The packet identifier
- **Sequence_Count** - Packet Sequence Count
- **Buffer_Length** - The number of bytes used in the packet buffer

## Packet_Id.T:

A packed record which holds a packet identifier.

Table 69: Packet_Id Packed Record : 16 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Packet_Types. Packet_Id | 0 to 65535 | 16 | 0 | 15 |

Field Descriptions:
- **Id** - The packet identifier

## Packet_Period.T:

A packed record which holds a packet identifier and period.

Table 70: Packet_Period Packed Record : 48 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Packet_Types. Packet_Id | 0 to 65535 | 16 | 0 | 15 |
| Period | Natural | 0 to 2147483647 | 32 | 16 | 47 |

Field Descriptions:
- **Id** - The packet identifier
- **Period** - The packet period, in ticks

## Parameter.T:

Generic parameter packet for holding a generic parameter

Table 71: Parameter Packed Record : 280 bits *(maximum)*

| Name | Type | Range | Size (Bits) | Start Bit | End Bit | Variable Length |
|---|---|---|---|---|---|---|
| Header | Parameter_ Header.T | - | 24 | 0 | 23 | - |
| Buffer | Parameter_ Types. Parameter_ Buffer_Type | - | 256 | 24 | 279 | Header.Buffer_ Length |

Field Descriptions:
- **Header** - The parameter header
- **Buffer** - A buffer to that contains the parameter type

## Parameter_Header.T:

Generic parameter header for holding arbitrary parameters

Table 72: Parameter_Header Packed Record : 24 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Parameter_Types.<br>Parameter_Id | 0 to 65535 | 16 | 0 | 15 |
| Buffer_Length | Parameter_Types.<br>Parameter_Buffer_<br>Length_Type | 0 to 32 | 8 | 16 | 23 |

Field Descriptions:

- **Id** - The parameter identifier
- **Buffer_Length** - The number of bytes used in the parameter type buffer

## Parameter_Update.T:

A record intended to be used as a provide/modify connector type for updating/fetching parameters.

Table 73: Parameter_Update Packed Record : 296 bits *(maximum)*

| Name | Type | Range | Size (Bits) | Start Bit | End Bit | Variable Length |
|---|---|---|---|---|---|---|
| Operation | Parameter_<br>Enums.<br>Parameter_<br>Operation_<br>Type.E | 0 => Stage<br>1 => Update<br>2 => Fetch | 8 | 0 | 7 | - |
| Status | Parameter_<br>Enums.<br>Parameter_<br>Update_<br>Status.E | 0 => Success<br>1 => Id_Error<br>2 => Validation_Error<br>3 => Length_Error | 8 | 8 | 15 | - |
| Param | Parameter.<br>T | - | 280 | 16 | 295 | - |

Field Descriptions:

- **Operation** - The parameter operation to perform.
- **Status** - The parameter return status.
- **Param** - The parameter that has been updated or fetched.

## Pet.T:

The pet datatype is used for servicing a watchdog. Included in this type is a count.

Table 74: Pet Packed Record : 32 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Count | Interfaces.<br>Unsigned_32 | 0 to 4294967295 | 32 | 0 | 31 |

Field Descriptions:

- **Count** - The cycle number of the pet.

## Pico_Example_Cpu_Monitor_Packet_Type.T:

This is the autocoded packet type for the CPU monitor component. It contains CPU utilization information for every task in the assembly.

Table 75: Pico_Example_Cpu_Monitor_Packet_Type Packed Record : 192 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Fault_Correction_ Instance_Active_ Usage_1 | Basic_Types.Byte | 0 to 255 | 8 | 0 | 7 |
| Fault_Correction_ Instance_Active_ Usage_2 | Basic_Types.Byte | 0 to 255 | 8 | 8 | 15 |
| Fault_Correction_ Instance_Active_ Usage_3 | Basic_Types.Byte | 0 to 255 | 8 | 16 | 23 |
| Ticker_Instance_ Active_Usage_1 | Basic_Types.Byte | 0 to 255 | 8 | 24 | 31 |
| Ticker_Instance_ Active_Usage_2 | Basic_Types.Byte | 0 to 255 | 8 | 32 | 39 |
| Ticker_Instance_ Active_Usage_3 | Basic_Types.Byte | 0 to 255 | 8 | 40 | 47 |
| Watchdog_Rate_Group_ Active_Usage_1 | Basic_Types.Byte | 0 to 255 | 8 | 48 | 55 |
| Watchdog_Rate_Group_ Active_Usage_2 | Basic_Types.Byte | 0 to 255 | 8 | 56 | 63 |
| Watchdog_Rate_Group_ Active_Usage_3 | Basic_Types.Byte | 0 to 255 | 8 | 64 | 71 |
| Slow_Rate_Group_ Active_Usage_1 | Basic_Types.Byte | 0 to 255 | 8 | 72 | 79 |
| Slow_Rate_Group_ Active_Usage_2 | Basic_Types.Byte | 0 to 255 | 8 | 80 | 87 |
| Slow_Rate_Group_ Active_Usage_3 | Basic_Types.Byte | 0 to 255 | 8 | 88 | 95 |
| Fast_Rate_Group_ Active_Usage_1 | Basic_Types.Byte | 0 to 255 | 8 | 96 | 103 |
| Fast_Rate_Group_ Active_Usage_2 | Basic_Types.Byte | 0 to 255 | 8 | 104 | 111 |
| Fast_Rate_Group_ Active_Usage_3 | Basic_Types.Byte | 0 to 255 | 8 | 112 | 119 |
| Command_Router_ Instance_Active_ Usage_1 | Basic_Types.Byte | 0 to 255 | 8 | 120 | 127 |
| Command_Router_ Instance_Active_ Usage_2 | Basic_Types.Byte | 0 to 255 | 8 | 128 | 135 |
| Command_Router_ Instance_Active_ Usage_3 | Basic_Types.Byte | 0 to 255 | 8 | 136 | 143 |
| Ccsds_Serial_ Interface_Instance_ Active_Usage_1 | Basic_Types.Byte | 0 to 255 | 8 | 144 | 151 |

| | | | | | |
|---|---|---|---|---|---|
| Ccsds_Serial_Interface_Instance_Active_Usage_2 | Basic_Types.Byte | 0 to 255 | 8 | 152 | 159 |
| Ccsds_Serial_Interface_Instance_Active_Usage_3 | Basic_Types.Byte | 0 to 255 | 8 | 160 | 167 |
| Ccsds_Serial_Interface_Instance_Listener_Usage_1 | Basic_Types.Byte | 0 to 255 | 8 | 168 | 175 |
| Ccsds_Serial_Interface_Instance_Listener_Usage_2 | Basic_Types.Byte | 0 to 255 | 8 | 176 | 183 |
| Ccsds_Serial_Interface_Instance_Listener_Usage_3 | Basic_Types.Byte | 0 to 255 | 8 | 184 | 191 |

Field Descriptions:

- **Fault_Correction_Instance_Active_Usage_1** - The period one CPU utilization percentage for the Fault_Correction_Instance.Active task.
- **Fault_Correction_Instance_Active_Usage_2** - The period two CPU utilization percentage for the Fault_Correction_Instance.Active task.
- **Fault_Correction_Instance_Active_Usage_3** - The period three CPU utilization percentage for the Fault_Correction_Instance.Active task.
- **Ticker_Instance_Active_Usage_1** - The period one CPU utilization percentage for the Ticker_Instance.Active task.
- **Ticker_Instance_Active_Usage_2** - The period two CPU utilization percentage for the Ticker_Instance.Active task.
- **Ticker_Instance_Active_Usage_3** - The period three CPU utilization percentage for the Ticker_Instance.Active task.
- **Watchdog_Rate_Group_Active_Usage_1** - The period one CPU utilization percentage for the Watchdog_Rate_Group.Active task.
- **Watchdog_Rate_Group_Active_Usage_2** - The period two CPU utilization percentage for the Watchdog_Rate_Group.Active task.
- **Watchdog_Rate_Group_Active_Usage_3** - The period three CPU utilization percentage for the Watchdog_Rate_Group.Active task.
- **Slow_Rate_Group_Active_Usage_1** - The period one CPU utilization percentage for the Slow_Rate_Group.Active task.
- **Slow_Rate_Group_Active_Usage_2** - The period two CPU utilization percentage for the Slow_Rate_Group.Active task.
- **Slow_Rate_Group_Active_Usage_3** - The period three CPU utilization percentage for the Slow_Rate_Group.Active task.
- **Fast_Rate_Group_Active_Usage_1** - The period one CPU utilization percentage for the Fast_Rate_Group.Active task.
- **Fast_Rate_Group_Active_Usage_2** - The period two CPU utilization percentage for the Fast_Rate_Group.Active task.
- **Fast_Rate_Group_Active_Usage_3** - The period three CPU utilization percentage for the Fast_Rate_Group.Active task.
- **Command_Router_Instance_Active_Usage_1** - The period one CPU utilization percentage for the Command_Router_Instance.Active task.

- **Command_Router_Instance_Active_Usage_2** - The period two CPU utilization percentage for the Command_Router_Instance.Active task.
- **Command_Router_Instance_Active_Usage_3** - The period three CPU utilization percentage for the Command_Router_Instance.Active task.
- **Ccsds_Serial_Interface_Instance_Active_Usage_1** - The period one CPU utilization percentage for the Ccsds_Serial_Interface_Instance.Active task.
- **Ccsds_Serial_Interface_Instance_Active_Usage_2** - The period two CPU utilization percentage for the Ccsds_Serial_Interface_Instance.Active task.
- **Ccsds_Serial_Interface_Instance_Active_Usage_3** - The period three CPU utilization percentage for the Ccsds_Serial_Interface_Instance.Active task.
- **Ccsds_Serial_Interface_Instance_Listener_Usage_1** - The period one CPU utilization percentage for the Ccsds_Serial_Interface_Instance.Listener task.
- **Ccsds_Serial_Interface_Instance_Listener_Usage_2** - The period two CPU utilization percentage for the Ccsds_Serial_Interface_Instance.Listener task.
- **Ccsds_Serial_Interface_Instance_Listener_Usage_3** - The period three CPU utilization percentage for the Ccsds_Serial_Interface_Instance.Listener task.

## Pico_Example_Fault_Responses_Packed_Id_Type.T:

This is an autocoded packed ID type for the fault correction component based on the fault IDs that it handles.

Table 76: Pico_Example_Fault_Responses_Packed_Id_Type Packed Record : 16 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Id | Pico_Example_Fault_Responses_Enums.Fault_Type.E | 0 => None<br>1 => Task_Watchdog_Instance_Slow_Rate_Group_Fault<br>2 => Task_Watchdog_Instance_Fast_Rate_Group_Fault<br>3 => Fault_Producer_Instance_Fault_1<br>4 => Fault_Producer_Instance_Fault_2 | 16 | 0 | 15 |

Field Descriptions:
- **Id** - The fault type identifier.

## Pico_Example_Fault_Responses_Status_Record.T:

This is an autocoded data product status record type for a Fault Correction component.

Table 77: Pico_Example_Fault_Responses_Status_Record Packed Record : 8 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Task_Watchdog_Instance_Slow_Rate_Group_Fault_Status | Fault_Correction_Enums.Status_Type.E | 0 => Disabled<br>1 => Nominal<br>2 => Fault_Detected<br>3 => Fault_Latched | 2 | 0 | 1 |

| | | | | | |
|---|---|---|---|---|---|
| Task_Watchdog_Instance_Fast_Rate_Group_Fault_Status | Fault_Correction_Enums.Status_Type.E | 0 => Disabled<br>1 => Nominal<br>2 => Fault_Detected<br>3 => Fault_Latched | 2 | 2 | 3 |
| Fault_Producer_Instance_Fault_1_Status | Fault_Correction_Enums.Status_Type.E | 0 => Disabled<br>1 => Nominal<br>2 => Fault_Detected<br>3 => Fault_Latched | 2 | 4 | 5 |
| Fault_Producer_Instance_Fault_2_Status | Fault_Correction_Enums.Status_Type.E | 0 => Disabled<br>1 => Nominal<br>2 => Fault_Detected<br>3 => Fault_Latched | 2 | 6 | 7 |

Field Descriptions:

- **Task_Watchdog_Instance_Slow_Rate_Group_Fault_Status** - The current status of the Task_Watchdog_Instance.Slow_Rate_Group_Fault fault response.
- **Task_Watchdog_Instance_Fast_Rate_Group_Fault_Status** - The current status of the Task_Watchdog_Instance.Fast_Rate_Group_Fault fault response.
- **Fault_Producer_Instance_Fault_1_Status** - The current status of the Fault_Producer_Instance.Fault_1 fault response.
- **Fault_Producer_Instance_Fault_2_Status** - The current status of the Fault_Producer_Instance.Fault_2 fault response.

## Pico_Example_Queue_Monitor_Packet_Type.T:

This is the autocoded packet type for the Queue Monitor component. It contains queue utilization information for queued component in the assembly.

Table 78: Pico_Example_Queue_Monitor_Packet_Type Packed Record : 160 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Slow_Rate_Group_Current_Usage | Basic_Types.Byte | 0 to 255 | 8 | 0 | 7 |
| Slow_Rate_Group_Maximum_Usage | Basic_Types.Byte | 0 to 255 | 8 | 8 | 15 |
| Fast_Rate_Group_Current_Usage | Basic_Types.Byte | 0 to 255 | 8 | 16 | 23 |
| Fast_Rate_Group_Maximum_Usage | Basic_Types.Byte | 0 to 255 | 8 | 24 | 31 |
| Watchdog_Rate_Group_Current_Usage | Basic_Types.Byte | 0 to 255 | 8 | 32 | 39 |
| Watchdog_Rate_Group_Maximum_Usage | Basic_Types.Byte | 0 to 255 | 8 | 40 | 47 |
| Command_Router_Instance_Current_Usage | Basic_Types.Byte | 0 to 255 | 8 | 48 | 55 |

| | | | | | |
|---|---|---|---|---|---|
| Command_Router_Instance_Maximum_Usage | Basic_Types.Byte | 0 to 255 | 8 | 56 | 63 |
| Product_Packetizer_Instance_Current_Usage | Basic_Types.Byte | 0 to 255 | 8 | 64 | 71 |
| Product_Packetizer_Instance_Maximum_Usage | Basic_Types.Byte | 0 to 255 | 8 | 72 | 79 |
| Ccsds_Serial_Interface_Instance_Current_Usage | Basic_Types.Byte | 0 to 255 | 8 | 80 | 87 |
| Ccsds_Serial_Interface_Instance_Maximum_Usage | Basic_Types.Byte | 0 to 255 | 8 | 88 | 95 |
| Counter_Instance_Current_Usage | Basic_Types.Byte | 0 to 255 | 8 | 96 | 103 |
| Counter_Instance_Maximum_Usage | Basic_Types.Byte | 0 to 255 | 8 | 104 | 111 |
| Oscillator_A_Current_Usage | Basic_Types.Byte | 0 to 255 | 8 | 112 | 119 |
| Oscillator_A_Maximum_Usage | Basic_Types.Byte | 0 to 255 | 8 | 120 | 127 |
| Oscillator_B_Current_Usage | Basic_Types.Byte | 0 to 255 | 8 | 128 | 135 |
| Oscillator_B_Maximum_Usage | Basic_Types.Byte | 0 to 255 | 8 | 136 | 143 |
| Fault_Correction_Instance_Current_Usage | Basic_Types.Byte | 0 to 255 | 8 | 144 | 151 |
| Fault_Correction_Instance_Maximum_Usage | Basic_Types.Byte | 0 to 255 | 8 | 152 | 159 |

Field Descriptions:

- **Slow_Rate_Group_Current_Usage** - The current percent usage of the Slow_Rate_Group internal queue.
- **Slow_Rate_Group_Maximum_Usage** - The maximum percent usage (high water mark) of the Slow_Rate_Group internal queue.
- **Fast_Rate_Group_Current_Usage** - The current percent usage of the Fast_Rate_Group internal queue.
- **Fast_Rate_Group_Maximum_Usage** - The maximum percent usage (high water mark) of the Fast_Rate_Group internal queue.
- **Watchdog_Rate_Group_Current_Usage** - The current percent usage of the Watchdog_Rate_Group internal queue.
- **Watchdog_Rate_Group_Maximum_Usage** - The maximum percent usage (high water mark) of the Watchdog_Rate_Group internal queue.
- **Command_Router_Instance_Current_Usage** - The current percent usage of the Command_Router_Instance internal queue.
- **Command_Router_Instance_Maximum_Usage** - The maximum percent usage (high water mark) of the Command_Router_Instance internal queue.
- **Product_Packetizer_Instance_Current_Usage** - The current percent usage of the

Product_Packetizer_Instance internal queue.

- **Product_Packetizer_Instance_Maximum_Usage** - The maximum percent usage (high water mark) of the Product_Packetizer_Instance internal queue.
- **Ccsds_Serial_Interface_Instance_Current_Usage** - The current percent usage of the Ccsds_Serial_Interface_Instance internal queue.
- **Ccsds_Serial_Interface_Instance_Maximum_Usage** - The maximum percent usage (high water mark) of the Ccsds_Serial_Interface_Instance internal queue.
- **Counter_Instance_Current_Usage** - The current percent usage of the Counter_Instance internal queue.
- **Counter_Instance_Maximum_Usage** - The maximum percent usage (high water mark) of the Counter_Instance internal queue.
- **Oscillator_A_Current_Usage** - The current percent usage of the Oscillator_A internal queue.
- **Oscillator_A_Maximum_Usage** - The maximum percent usage (high water mark) of the Oscillator_A internal queue.
- **Oscillator_B_Current_Usage** - The current percent usage of the Oscillator_B internal queue.
- **Oscillator_B_Maximum_Usage** - The maximum percent usage (high water mark) of the Oscillator_B internal queue.
- **Fault_Correction_Instance_Current_Usage** - The current percent usage of the Fault_Correction_Instance internal queue.
- **Fault_Correction_Instance_Maximum_Usage** - The maximum percent usage (high water mark) of the Fault_Correction_Instance internal queue.

## Pico_Example_Stack_Monitor_Packet_Type.T:

This is the autocoded packet type for the stack monitor component. It contains stack and secondary stack utilization information for every task in the assembly.

Table 79: Pico_Example_Stack_Monitor_Packet_Type Packed Record : 128 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Fault_Correction_ Instance_Active_ Primary_Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 0 | 7 |
| Fault_Correction_ Instance_Active_ Secondary_Stack_ Usage | Basic_Types.Byte | 0 to 255 | 8 | 8 | 15 |
| Ticker_Instance_ Active_Primary_ Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 16 | 23 |
| Ticker_Instance_ Active_Secondary_ Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 24 | 31 |
| Watchdog_Rate_Group_ Active_Primary_ Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 32 | 39 |
| Watchdog_Rate_Group_ Active_Secondary_ Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 40 | 47 |

| | | | | | |
|---|---|---|---|---|---|
| Slow_Rate_Group_ Active_Primary_ Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 48 | 55 |
| Slow_Rate_Group_ Active_Secondary_ Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 56 | 63 |
| Fast_Rate_Group_ Active_Primary_ Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 64 | 71 |
| Fast_Rate_Group_ Active_Secondary_ Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 72 | 79 |
| Command_Router_ Instance_Active_ Primary_Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 80 | 87 |
| Command_Router_ Instance_Active_ Secondary_Stack_ Usage | Basic_Types.Byte | 0 to 255 | 8 | 88 | 95 |
| Ccsds_Serial_ Interface_Instance_ Active_Primary_ Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 96 | 103 |
| Ccsds_Serial_ Interface_Instance_ Active_Secondary_ Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 104 | 111 |
| Ccsds_Serial_ Interface_Instance_ Listener_Primary_ Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 112 | 119 |
| Ccsds_Serial_ Interface_Instance_ Listener_Secondary_ Stack_Usage | Basic_Types.Byte | 0 to 255 | 8 | 120 | 127 |

Field Descriptions:

- **Fault_Correction_Instance_Active_Primary_Stack_Usage** - The primary stack utilization percentage for the Fault_Correction_Instance.Active task.
- **Fault_Correction_Instance_Active_Secondary_Stack_Usage** - The secondary stack utilization percentage for the Fault_Correction_Instance.Active task.
- **Ticker_Instance_Active_Primary_Stack_Usage** - The primary stack utilization percentage for the Ticker_Instance.Active task.
- **Ticker_Instance_Active_Secondary_Stack_Usage** - The secondary stack utilization percentage for the Ticker_Instance.Active task.
- **Watchdog_Rate_Group_Active_Primary_Stack_Usage** - The primary stack utilization percentage for the Watchdog_Rate_Group.Active task.
- **Watchdog_Rate_Group_Active_Secondary_Stack_Usage** - The secondary stack utilization percentage for the Watchdog_Rate_Group.Active task.
- **Slow_Rate_Group_Active_Primary_Stack_Usage** - The primary stack utilization percentage for the Slow_Rate_Group.Active task.
- **Slow_Rate_Group_Active_Secondary_Stack_Usage** - The secondary stack utilization

percentage for the Slow_Rate_Group.Active task.

- **Fast_Rate_Group_Active_Primary_Stack_Usage** - The primary stack utilization percentage for the Fast_Rate_Group.Active task.
- **Fast_Rate_Group_Active_Secondary_Stack_Usage** - The secondary stack utilization percentage for the Fast_Rate_Group.Active task.
- **Command_Router_Instance_Active_Primary_Stack_Usage** - The primary stack utilization percentage for the Command_Router_Instance.Active task.
- **Command_Router_Instance_Active_Secondary_Stack_Usage** - The secondary stack utilization percentage for the Command_Router_Instance.Active task.
- **Ccsds_Serial_Interface_Instance_Active_Primary_Stack_Usage** - The primary stack utilization percentage for the Ccsds_Serial_Interface_Instance.Active task.
- **Ccsds_Serial_Interface_Instance_Active_Secondary_Stack_Usage** - The secondary stack utilization percentage for the Ccsds_Serial_Interface_Instance.Active task.
- **Ccsds_Serial_Interface_Instance_Listener_Primary_Stack_Usage** - The primary stack utilization percentage for the Ccsds_Serial_Interface_Instance.Listener task.
- **Ccsds_Serial_Interface_Instance_Listener_Secondary_Stack_Usage** - The secondary stack utilization percentage for the Ccsds_Serial_Interface_Instance.Listener task.

## Pico_Example_Task_Watchdog_List_State_Record.T:

This is an autocoded data product status record type for a Task Watchdog component regarding the status of checking each connector for pets.

Table 80: Pico_Example_Task_Watchdog_List_State_Record Packed Record : 8 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Slow_Rate_Group_State | Task_Watchdog_Enums.Watchdog_Action_State.E | 0 => Disabled<br>1 => Warn<br>2 => Error_Fault | 2 | 0 | 1 |
| Fast_Rate_Group_State | Task_Watchdog_Enums.Watchdog_Action_State.E | 0 => Disabled<br>1 => Warn<br>2 => Error_Fault | 2 | 2 | 3 |
| Reserved_0 | Task_Watchdog_Types.Two_Bit_Padding_Type | 0 to 3 | 2 | 4 | 5 |
| Reserved_1 | Task_Watchdog_Types.Two_Bit_Padding_Type | 0 to 3 | 2 | 6 | 7 |

Field Descriptions:

- **Slow_Rate_Group_State** - The current state of the Slow_Rate_Group watchdog action.
- **Fast_Rate_Group_State** - The current state of the Fast_Rate_Group watchdog action.
- **Reserved_0** - Padding bits, not used.
- **Reserved_1** - Padding bits, not used.

## Pico_Example_Task_Watchdog_List_Watchdog_Action_Cmd.T:

This is an autocoded packed enumeration record which contains an enumeration literal for each task that the task watchdog manages. This record contains information for changing the limit of a

specific watchdog connector via this enumeration.

*Preamble (inline Ada definitions):*

```
type Task_Enum_Type is (
  Slow_Rate_Group,
  Fast_Rate_Group
);
for Task_Enum_Type use (
  Slow_Rate_Group => 1,
  Fast_Rate_Group => 2
);
```

Table 81: Pico_Example_Task_Watchdog_List_Watchdog_Action_Cmd Packed Record : 24 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Watchdog_Task | Task_Enum_Type | 1 => Slow_Rate_Group<br>2 => Fast_Rate_Group | 16 | 0 | 15 |
| New_Action | Task_Watchdog_Enums.Watchdog_Action_State.E | 0 => Disabled<br>1 => Warn<br>2 => Error_Fault | 8 | 16 | 23 |

Field Descriptions:

- **Watchdog_Task** - The task watchdog task enumeration.
- **New_Action** - The new value of the action for the specific associated connector

## Pico_Example_Task_Watchdog_List_Watchdog_Limit_Cmd.T:

This is an autocoded packed enumeration record which contains an enumeration literal for each task that the task watchdog manages. This record contains information for changing the limit of a specific watchdog connector via this enumeration.

*Preamble (inline Ada definitions):*

```
type Task_Enum_Type is (
  Slow_Rate_Group,
  Fast_Rate_Group
);
for Task_Enum_Type use (
  Slow_Rate_Group => 1,
  Fast_Rate_Group => 2
);
```

Table 82: Pico_Example_Task_Watchdog_List_Watchdog_Limit_Cmd Packed Record : 32 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Watchdog_Task | Task_Enum_Type | 1 => Slow_Rate_Group<br>2 => Fast_Rate_Group | 16 | 0 | 15 |
| New_Limit | Task_Watchdog_Types.Missed_Pet_Limit_Type | 1 to 65534 | 16 | 16 | 31 |

Field Descriptions:

- **Watchdog_Task** - The task watchdog task enumeration.
- **New_Limit** - The new value of the limit for the specific associated connector

## Stack_Trace_Addresses.T:

An array of packed addresses in big endian. This is sized to easily fit a normal stack trace.

Table 83: Stack_Trace_Addresses Packed Array : 3200 bits

| Type | Range | Element Size (Bits) | Length | Total Size (Bits) |
|---|---|---|---|---|
| **Packed_Address.T** | - | 32 | 100 | 3200 |

## Sys_Time.T:

A record which holds a time stamp using GPS format including seconds and subseconds since epoch (1-5-1980 to 1-6-1980 midnight).

Table 84: Sys_Time Packed Record : 64 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Seconds | Interfaces. Unsigned_32 | 0 to 4294967295 | 32 | 0 | 31 |
| Subseconds | Interfaces. Unsigned_32 | 0 to 4294967295 | 32 | 32 | 63 |

Field Descriptions:

- **Seconds** - The number of seconds elapsed since epoch.
- **Subseconds** - The number of 1/(2^32) sub-seconds.

## Task_Timing_Report.T:

Record which holds timing reports for the all-time maximum and recent maximum of an executing task

Table 85: Task_Timing_Report Packed Record : 256 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Max | Timing_Report.T | - | 128 | 0 | 127 |
| Recent_Max | Timing_Report.T | - | 128 | 128 | 255 |

Field Descriptions:

- **Max** - The maximum recorded timing report since start up.
- **Recent_Max** - The maximum recorded timing report during some recent time invterval.

## Td_Full_Queue_Param.T:

This is a type that contains useful information about a component full queue.

Table 86: Td_Full_Queue_Param Packed Record : 112 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Dropped_Tick | Tick.T | - | 96 | 0 | 95 |
| Index | Connector_Types. Connector_Index_ Type | 1 to 65535 | 16 | 96 | 111 |

Field Descriptions:

- **Dropped_Tick** - The tick during which the component's queue was found to be full.
- **Index** - The tick divider index number of the component that had a full queue.

## Tick.T:

The tick datatype used for periodic scheduling. Included in this type is the Time associated with a tick and a count.

Table 87: Tick Packed Record : 96 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Time | Sys_Time.T | - | 64 | 0 | 63 |
| Count | Interfaces. Unsigned_32 | 0 to 4294967295 | 32 | 64 | 95 |

Field Descriptions:

- **Time** - The timestamp associated with the tick.
- **Count** - The cycle number of the tick.

## Time_Exceeded.T:

Datatype used for reporting time deltas at a particular count number.

Table 88: Time_Exceeded Packed Record : 96 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Time_Delta | Delta_Time.T | - | 64 | 0 | 63 |
| Count | Interfaces. Unsigned_32 | 0 to 4294967295 | 32 | 64 | 95 |

Field Descriptions:

- **Time_Delta** - The time delta.
- **Count** - The cycle number of the tick.

## Timing_Report.T:

Record which holds wall time and execution time to describe the runtime performance some piece of code.

Table 89: Timing_Report Packed Record : 128 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Wall_Time | Delta_Time.T | - | 64 | 0 | 63 |
| Execution_Time | Delta_Time.T | - | 64 | 64 | 127 |

Field Descriptions:

- **Wall_Time** - The wall time associated with the measurement.
- **Execution_Time** - The time the task spent executing on the CPU.

### Watchdog_Action_Cmd.T:

This record contains information for changing the limit of a specific watchdog connector.

Table 90: Watchdog_Action_Cmd Packed Record : 24 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Index | Connector_Types.Connector_Index_Type | 1 to 65535 | 16 | 0 | 15 |
| New_Action | Task_Watchdog_Enums.Watchdog_Action_State.E | 0 => Disabled<br>1 => Warn<br>2 => Error_Fault | 8 | 16 | 23 |

Field Descriptions:

- **Index** - The index of the connector that the command wants to change the limit of
- **New_Action** - The new value of the action for the specific associated connector

### Watchdog_Limit_Cmd.T:

This record contains information for changing the limit of a specific watchdog connector.

Table 91: Watchdog_Limit_Cmd Packed Record : 32 bits

| Name | Type | Range | Size (Bits) | Start Bit | End Bit |
|---|---|---|---|---|---|
| Index | Connector_Types.Connector_Index_Type | 1 to 65535 | 16 | 0 | 15 |
| New_Limit | Task_Watchdog_Types.Missed_Pet_Limit_Type | 1 to 65534 | 16 | 16 | 31 |

Field Descriptions:

- **Index** - The index of the connector that the command wants to change the limit of
- **New_Limit** - The new value of the limit for the specific associated connector

## 3.3 Enumerations

The following section outlines any enumerations used in the assembly.

## Basic_Enums.Enable_Disable_Type.E:

This enumeration includes enable and disable state.

Table 92: Enable_Disable_Type Literals:

| Name | Value | Description |
|---|---|---|
| Disabled | 0 | The state is disabled. |
| Enabled | 1 | The state is enabled. |

## Ccsds_Enums.Ccsds_Packet_Type.E:

This single bit is used to identify that this is a Telecommand Packet or a Telemetry Packet. A Telemetry Packet has this bit set to value 0; therefore, for all Telecommand Packets Bit 3 shall be set to value 1.

Table 93: Ccsds_Packet_Type Literals:

| Name | Value | Description |
|---|---|---|
| Telemetry | 0 | Indicates a telemetry packet |
| Telecommand | 1 | Indicates a telecommand packet |

## Ccsds_Enums.Ccsds_Secondary_Header_Indicator.E:

This one bit flag signals the presence (Bit 4 = 1) or absence (Bit 4 = 0) of a Secondary Header data structure within the packet.

Table 94: Ccsds_Secondary_Header_Indicator Literals:

| Name | Value | Description |
|---|---|---|
| Secondary_Header_Not_Present | 0 | Indicates that the secondary header is not present within the packet |
| Secondary_Header_Present | 1 | Indicates that the secondary header is present within the packet |

## Ccsds_Enums.Ccsds_Sequence_Flag.E:

This flag provides a method for defining whether this packet is a first, last, or intermediate component of a higher layer data structure.

Table 95: Ccsds_Sequence_Flag Literals:

| Name | Value | Description |
|---|---|---|
| Continuationsegment | 0 | Continuation component of higher data structure |
| Firstsegment | 1 | First component of higher data structure |
| Lastsegment | 2 | Last component of higher data structure |
| Unsegmented | 3 | Standalone packet |

## Command_Enums.Command_Response_Status.E:

This status enumerations provides information on the success/failure of a command through the command response connector.

Table 96: Command_Response_Status Literals:

| Name | Value | Description |
|---|---|---|
| Success | 0 | Command was passed to the handler and successfully executed. |
| Failure | 1 | Command was passed to the handler not successfully executed. |
| Id_Error | 2 | Command id was not valid. |
| Validation_Error | 3 | Command parameters were not successfully validated. |
| Length_Error | 4 | Command length was not correct. |
| Dropped | 5 | Command overflowed a component queue and was dropped. |
| Register | 6 | This status is used to register a command with the command routing system. |
| Register_Source | 7 | This status is used to register command sender's source id with the command router for command response forwarding. |

## Data_Product_Enums.Fetch_Status.E:

This status denotes whether a data product fetch was successful.

Table 97: Fetch_Status Literals:

| Name | Value | Description |
|---|---|---|
| Success | 0 | The data product was returned successfully. |
| Not_Available | 1 | No data product is yet available for the provided id. |
| Id_Out_Of_Range | 2 | The data product id was out of range. |

## Event_Filter_Entry_Enums.Global_Filter_State.E:

This is an enumeration indicating if the filter component is enabled/disabled

Table 98: Global_Filter_State Literals:

| Name | Value | Description |
|---|---|---|
| Disabled | 0 | Individual event states will be ignored and no events will be filtered |
| Enabled | 1 | Individual event states will be used to determine if the event needs to be filtered |

## Event_Filter_Enums.Issue_Packet_Type.E:

An enumeration for commands once the state is change for ground to determine if they want to send the state packet or not

Table 99: Issue_Packet_Type Literals:

| Name | Value | Description |
|---|---|---|
| No_Issue | 0 | Packet will not be issued |
| Issue | 1 | Packet will be issued |

## Event_Limiter_Enums.Issue_Packet_Type.E:

An enumeration for commands once the state is change for ground to determine if they want to send the state packet or not

Table 100: Issue_Packet_Type Literals:

| Name | Value | Description |
|---|---|---|
| No_Issue | 0 | Packet will not be issued |
| Issue | 1 | Packet will be issued |

## Fault_Correction_Enums.Status_Type.E:

This status enumerations provides the status of a fault response.

Table 101: Status_Type Literals:

| Name | Value | Description |
|---|---|---|
| Disabled | 0 | The fault response is disabled. |
| Nominal | 1 | The fault response is enabled and no fault has been detected. |
| Fault_Detected | 2 | A fault has been detected and a command response has been issued. |
| Fault_Latched | 3 | A fault has been detected and a command response has been issued. The fualt has been latched, so any more faults received with this ID will not issue another response until the this fault response has been unlatched. |

## Parameter_Enums.Parameter_Operation_Type.E:

This enumeration lists the different parameter operations that can be performed.

Table 102: Parameter_Operation_Type Literals:

| Name | Value | Description |
|---|---|---|
| Stage | 0 | Stage the parameter. |
| Update | 1 | All parameters are staged, it is ok to update all parameters now. |
| Fetch | 2 | Fetch the parameter. |

## Parameter_Enums.Parameter_Update_Status.E:

This status enumerations provides information on the success/failure of a parameter operation.

Table 103: Parameter_Update_Status Literals:

| Name | Value | Description |
|---|---|---|
| Success | 0 | Parameter was successfully staged. |
| Id_Error | 1 | Parameter id was not valid. |
| Validation_Error | 2 | Parameter values were not successfully validated. |
| Length_Error | 3 | Parameter length was not correct. |

## Pico_Example_Fault_Responses_Enums.Fault_Type.E:

This enumerations lists all the possible fault conditions possible in the assembly. The value of the enumeration corresponds to the fault identifier for that fault.

Table 104: Fault_Type Literals:

| Name | Value | Description |
|---|---|---|
| None | 0 | No fault present. |
| Task_Watchdog_Instance_Slow_Rate_Group_Fault | 1 | Throw noop with value 1 if this fault occurs. |
| Task_Watchdog_Instance_Fast_Rate_Group_Fault | 2 | Throw noop with value 2 if this fault occurs. |
| Fault_Producer_Instance_Fault_1 | 3 | Throw noop with value 3 if this fault occurs. |
| Fault_Producer_Instance_Fault_2 | 4 | Throw noop with value 4 if this fault occurs. |

## Task_Watchdog_Enums.Watchdog_Action_State.E:

The state for if each watchdog for which action it should take.

Table 105: Watchdog_Action_State Literals:

| Name | Value | Description |
|---|---|---|
| Disabled | 0 | |
| Warn | 1 | |
| Error_Fault | 2 | |

## Task_Watchdog_Enums.Watchdog_Enabled_State.E:

The state for if each watchdog is enabled or disabled for checking

Table 106: Watchdog_Enabled_State Literals:

| Name | Value | Description |
|---|---|---|
| Disabled | 0 | |
| Enabled | 1 | |

## Two_Counter_Entry_Enums.Event_State_Type.E:

This is a single bit identifying if the event limiter is enabled for a particular ID.

Table 107: Event_State_Type Literals:

| Name | Value | Description |
|---|---|---|
| Disabled | 0 | Event will not be limited |
| Enabled | 1 | Event will be limited |