Level 2

Main

- Create an ECU Instance
- Initialize ECU
- Register Service Availability Handler

ECU - __init__

Create instance for Vsomeip Runtime

Vsomeip Runtime - __init__

From bindings, runtime->get()

ECU - Initialize

Create instance for ConfigManager with json config file path and ECU name

Create instance for VsomeipApp with vsomeip config file and VsomeipRuntime

Initialize all provided and required Services

Register all provided and required Services

ConfigManager - __init__

self._vsomeip_config_file stores path to vsomeip config file

Service - __init__

Initialize all Event Groups, Event,Fields,Methods

EventGroup/Event/Field/Method - __init__

Register itself to owner Service

VsomeipApp - offer_service()

app.offer_service() with service_id,instance_id,major and minor

VsomeipApp.offer_event()

app.offer_event()

Provided Event.offer()

Event.notify()

payload = bytes (self._value)

Left column (flowchart nodes)

- Register Event and Field Notification Handler
- Register Method and Field Message Handler
- Start ECU
- Set data value for Event and Field and Notify
- Set data value for Method and send

VsomeipApp - start_app

- app = create_application from bindings with app name and vsomeip config file
- app.init()
- app.start()

Provided Service- start

- vsomeipApp.offer_service()
- offer provided Events
- offer provided Fields
- offer provided Event Groups
- offer provided Methods

Provided Field.offer()

- VsomeipApp.offer_event()
- Register setter message handler
- Register getter message handler
- Field.notify()

VsomeipApp.offer_event()

- Event.notify()

VsomeipApp.notify(payload)

VsomeipApp.register_message_handler_event()

- app.register_message_handler()

Event.notify()

- payload = bytes (self._value)
- VsomeipApp.notify(payload)

VsomeipApp.notify()

- VsomeipRuntime.create_payload()
- payload.set_raw_data()
- app.notify() with service_id,instance_id,event_id,payload

VsomeipApp.handle_subscription()

- app.register_subscription_handler()

Provided EventGroup.offer()

- VsomeipApp.handle_subscription()

VsomeipApp.register_message_handler_method()

- app.register_message_handler()

Provided Method.offer()

- VsomeipApp.register_message_handler()

vsomeipApp.request_service()

- app.request_service()

ECU- start

- Start VsomeipApp
- Start registered provided services
- Start registered required services

Required Service- start

- vsomeipApp.register_service_availability_handler()
- VsomeipApp.request_service()

vsomeipApp.register_service_availability_handler()

- app.register_availability_handler()

Required Event.subscribe()

- vsomeipApp.request_event()
- For all eventGroups, VsomeipApp.subscribe_to_eventgroup()

vsomeipApp.request_event()

- app.request_event()

Required Field.subscribe()

- vsomeipApp.request_field

vsomeipApp.subscribe_to_eventgroup()

- app.subscribe()

Stop ECU

subscribe to required Events

For all eventGroups, VsomeipApp.subscribe_to_eventgroup()

Subscribe to required Fields

For all eventGroups, VsomeipApp.register_subscription_status_handler()

VsomeipApp.register_message_handler()

For all eventGroups, VsomeipApp.register_subscription_status_handler()

app.register_subscription_status_handler()

VsomeipApp.register_message_handler()

vsomeipApp.register_subscription_status_handler()

app.register_subscription_status_handler()

vsomeipApp.request_field()

app.request_event()