

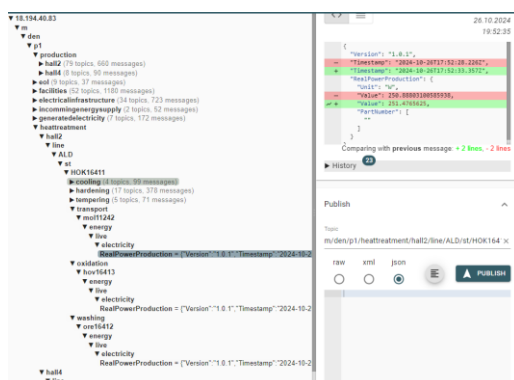
# Aufgabe Rimma

## Hintergrund:

SyncTwin erstellt in einem aktuellen Projekt eine Omniverse KIT App für die 3D Visualisierung von Maschinenzuständen in Omniverse mit OpenUSD und Omni.UI:

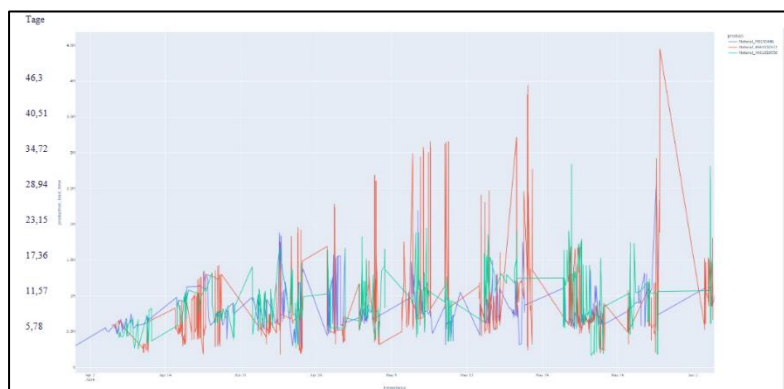


Die Maschinenzustände wie Status und aktueller Energieverbrauch werden über einen MQTT Service übermittelt:



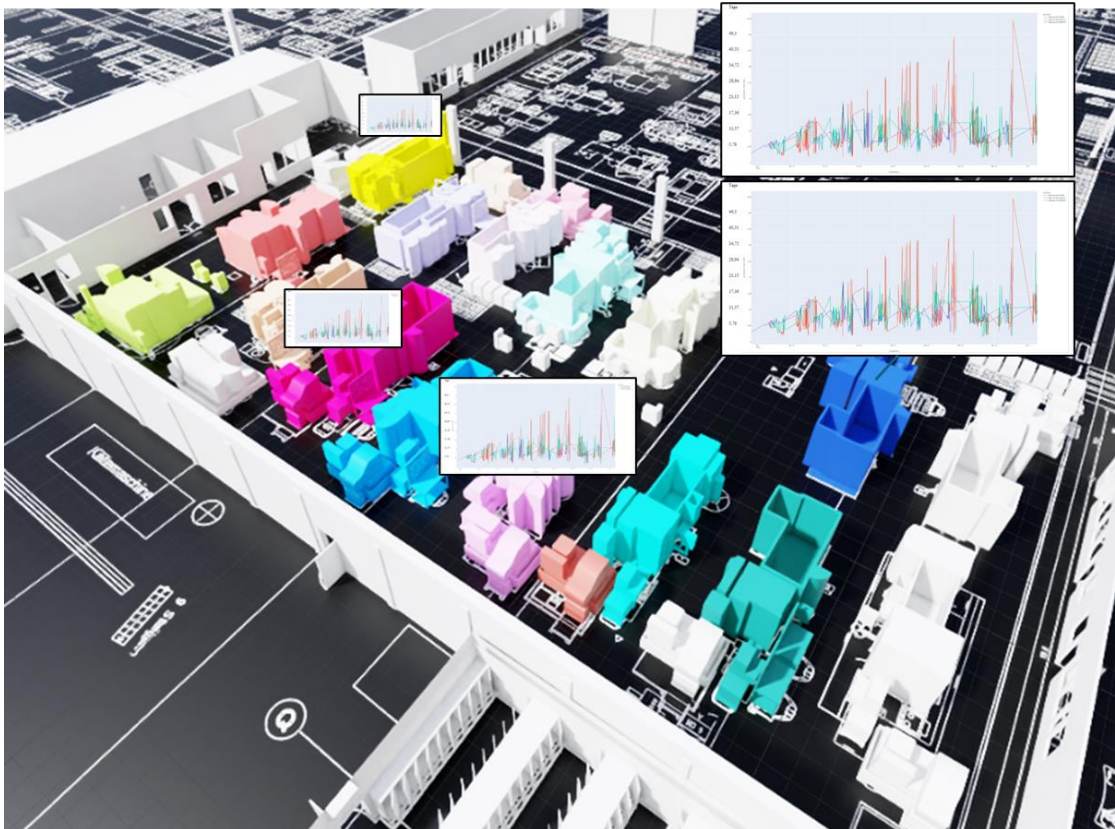
Der Zustand der Maschine (z.B. Fehler oder Laden etc) wird über eine entsprechende Einfärbung (siehe oben) gekennzeichnet.

Andere Daten wie z.B. der Verlaufs der Strom-Verbrauchsdaten wollen wir mit einem entsprechenden Graph mit den letzten x werten anzeigen.



Wir wollen die Graphen als Overlay in der SyncTwin Omniverse KIT App anzeigen, dabei soll der Graph über der betreffenden Maschine angezeigt werden. Hierbei wollen wir zwei Anzeigearten umsetzen – a) Fixe Größe im Vordergrund oder b) Größe abhängig von der Entfernung zur Kamera.

In beiden Fällen sollen die Graphen immer parallel zur Bildebene dargestellt werden (Billboarding)



## Aufgabenstellung:

1. Erstelle eine Klassenstruktur die Daten aus MQTT Topics entgegen nehmen und die letzten x Werte verwalten kann (Mqtt Client wird bereitgestellt)
2. Erstelle eine initiale Darstellung der Daten als Plot mit matplotlib
3. Vollziehe das Tutorial für Widget Overlays nach:  
[https://github.com/NVIDIA-Omniverse/kit-extension-sample-ui-scene/tree/main/exts/omni.example.ui\\_scene.widget\\_info](https://github.com/NVIDIA-Omniverse/kit-extension-sample-ui-scene/tree/main/exts/omni.example.ui_scene.widget_info)
4. Erweitere die KIT-Extension zur Darstellung der Daten aus 1.) mit der Plot UI Komponente  
<https://docs.omniverse.nvidia.com/kit/docs/omni.ui/latest/omni.ui/omni.ui.Plot.html>

## Hintergrundinfos:

OpenUSD

<https://docs.omniverse.nvidia.com/usd/latest/index.html>

MQTT Artikel

[How to control a digital twin asset with MQTT in NVIDIA Omniverse | by Michael T. Wagner | Medium](#)

Extensions

[Getting Started with Extensions — Omniverse Kit 106.2.0 documentation](#)