Runtime der WebPLC

# Ideen

* Programm besteht aus einer Liste mit FunctionBlockConfigurations und je einer Liste mit BoolVariableConfigurations, IntVariableConfigurations, FloatVariableConfigurations
  + Es gibt SimpleNInputMOutputConfigurations mit
    - FunctionCodeId
    - GUUID (wird beim Anlegen eines Blocks vergeben)
    - Liste der VariablenIds der Inputs
    - Liste der VariablenIDs der Output
    - Position der Repräsentanz auf der grafischen oberfläche
  + Für Blöcke, die eine weitergehende Konfiguration benötigen (insbesondere also Informationen, die zur Laufzeit nicht über „Inputs“ verändert werden können, gibt es eigene Konfigurationsobjekte.
* Initialisierung
  + Hardware-Spezifisch
  + Erstelle für jede FunctionBlockConfiguration eine FunctionBlockRuntime-Instanz (gemäß FunctionCodeId) und übergebe dieser eine Referenz auf ihre FunctionBlockConfiguration
  + Außerdem wird für jeden Typ VariableConfigurations ein entsprechend langes Array angelegt
  + Rufe von jeder FunctionBlockRuntime zunächst die InitStage1, dann InitStage2 und dann InitStage3 auf. In einem gesonderten Dokument ist zu definieren, was in diesen Stages zu passieren hat. Möglicherweise werden zukünftig noch mehr Stages benötigt. Wichtig ist, dass nach der dritten Stage alles betriebsbereit (also insbesondere seine Ausgnagsvariablen gesetzt hat) ist und es zwischendurch kein unaufgelösten Abhängigkeiten gibt. Dafür sollten drei Init-Stages eigentlich ausreichend sein. Falls der Block in der Lage ist, seinen Zustand aus dem GUUID-Speicher wieder herzustellen, sollte er das tun.
  + Rufe von jedem Blockdie Process-Methode auf und übergebe ihr ein RuntimeContext-Objekt dieses kann
    - Werte von Variablen lesen und schreiben (kein Schutz vor dem Lesen und Beschreiben „fremder“ Variablen!!!)
    - MonotonicTime
    - WallclockTime
    - Werte von Variablen persistieren
  + Es könnte noch eine Funktion OnPowerLoss() geben, der Notfallmaßnahmen ergreift
  + Lesen der