Dokumentation MetaModel

Klasse AutosarSystem

Eigenschaften: -

Relationen: besteht aus Bricks, SWCs und Connections

Konfiguration: Diese Klasse dient als Basisklasse. in dieser Klasse befinden sich die beiden Bricks, die verschiedenen Softwarekomponenten und die Connections.

Klasse Brick

Eigenschaften: enthält den bluetoothMode und die bluetoothMAC

Relationen: sind Bestandteil des AutosarSystems, enthält Referenz zu SWCs, besteht aus HardwareConnections

Konfiguration: Das Projekt hat 2 Bricks. Jeder Brick hat Referenzen auf die Softwarekomponenten, die zu ihm gehören. Zudem besteht jeder Brick aus mehreren Hardware-Verbindungen.

Klasse SWC

Eigenschaften: -

Relationen: gehört zu einem Brick, ist Bestandteil des Autosarsystems, besteht aus Runnables und Ports

Konfiguration: Jede Softwarekomponente ist einem der beiden Bricks zugeordnet und besteht aus Runnables und Ports. Sie bilden die Tasks, die die übergeordnete Funktionalität managet.

Klasse HardwareConnection

Eigenschaften:

Relationen: ist Bestandteil eines Bricks, hat eine Referenz zu einem ECUPort, ist eine Generalisierung von ActuatorSensor

Konfiguration: in jeder Hardware Verbindung wird der zugehörige ECU-Port referenziert und bildet die Generalisierung für weitere Klassen in denen die Funktionalität näher festgelegt wird. Die nächste Spezialisierung ist ActuatorSensor.

Klasse ActuatorSensor

Eigenschaften: enthält die PortNr

Relationen: ist eine Spezialisierung von HardwareConnection, ist eine Generalisierung von RawAccess

Konfiguration: In der Klasse steht der am Brick verwendete Port und bildet die Generalisierung für weitere Klassen in denen die Funktionalität näher festgelegt wird. Die nächste Spezialisierung ist RawAccess

Klasse RawAccess

Eigenschaften: enthält den Access-Modus

Relationen: ist eine Spezialisierung von ActuatorSensor, ist eine Generalisierung von I2CExpander und ADC

Konfiguration: die Klasse enthält Informationen darüber, ob an dem Port Daten nur gelesen, nur geschrieben oder gelesen und geschrieben werden. Der Informationsaustausch geschieht entweder über einen I²C-Expander oder einen A/D-Konverter

Klasse ECRobotAccess

Eigenschaften: enthält den Access-Modus

Relationen: ist eine Spezialisierung von ActuatorSensor, ist eine Generalisierung von HardwareUltraschall und HardwareMotor

Konfiguration: Die Klasse ist der Klasse RawAccess ähnlich und ist für Peripherie, die die ECRobot API nutzen

Klasse I2CExpander

Eigenschaften: enthält die Adresse, den Modus und einen Pin

Relationen: ist eine Spezialisierung von RawAccess

Konfiguration: Die Klasse enthält Informationen über die Adresse des i²c-Expanders , den verwendeten Modus, der entweder DIO oder ADC sein kann, sowie den Pin.

Klasse Port

Eigenschaften:

Relationen: ist Bestandteil einer SWC, ist eine Generalisierung von ECUPort und SoftwarePort

Konfiguration: jeder Port gehört zu einer Softwarekomponente und kann entweder ein ECUPort oder ein Softwareport sein.

Klasse ECUPort

Eigenschaften:

Relationen: ist eine Generalisierung von Motor, Taster, Ultraschall, LED, JoystickTaster, JoystickVertical und JoystickHorizontal

Konfiguration: an jedem ECUPort kann ein Motor, ein Sensor oder ein Bestandteil des Joysticks angeschlossen sein. Diese können Daten schreiben und/oder lesen. Die weitere Funktionalität ist in der Klasse HardwareConnection und ihren Unterklassen beschrieben.

Klasse Runnable

Eigenschaften:

Relationen: ist Bestandteil einer SWC, besteht aus einem TriggerEvent

Konfiguration: Jede Runnable ist Bestandteil einer SWC. Diese sind für die einzelnen speziellen Funktionalitäten zuständig und sind jeweils mit einem Event verknüpft, welches die Runnable startet, wenn es gesetzt wird.

Klasse Triggerevent

Eigenschaften:

Relationen: gehört zu einer Runnable, ist eine Generalisierung von TimeTrigger und TriggerPortTrigger

Konfiguration: Events können auf verschiedene Weise gesetzt werden und aktivieren jeweils eine Runnable

Klasse TimeTrigger

Eigenschaften: enthält die Zykluszeit in Millisekunden

Relationen: ist eine Spezialisierung von TriggerEvent

Konfiguration: Die Klasse beschreibt ein Event und einen damit verknüpften, automatisch gestartetem Alarm, welcher das Event nach einer bestimmten Zykluszeit immer wieder setzt und die verknüpfte Runnable somit zyklisch immer wieder aufruft.

Klasse TriggerPortTrigger

Eigenschaften:

Relationen: ist eine Spezialisierung von Triggerevent, hat eine Referenz zu einem Triggerport

Konfiguration: Die Klasse beschreibt ein Event, welches durch einen Triggerport gesetzt wird.

Klasse Connection

Eigenschaften:

Relationen: Referenz zu einem Softwareport als Input und zu einem Softwareport als Output

Konfiguration: stellen die Verbindungen zwischen den Softwarekomponenten dar. Jede Connection verbindet zwei Softwareports, von denen einer den Output der einen SWC und der andere den Input einer weiteren SWC bildet.

Klasse Softwareport

Eigenschaften:

Relationen: ist eine Generalisierung von SenderReceiverPort, TriggerPor und ServerClientPort

Konfiguration: Die Klasse ist die Oberklasse für die verschiedenen Arten von Softwareports. Diese bilden die Schnittstelle zwischen den SWCs, im Gegensatz zu den ECU-Ports, die die Schnittstelle zwischen einer SWC und einem Peripheriegerät bilden.

Klasse SenderReceiverPort

Eigenschaften: enthält den Typ

Relationen: ist eine Spezialisierung von Softwareport

Konfiguration: Über einen SenderReceiverPort werden Datenelemente von einer Softwarekomponente zu einer weiteren gesendet. Im Typ steht ob der Port Daten sendet oder empfängt.

Klasse TriggerPort

Eigenschaften: enthält den Typ

Relationen: ist eine Spezialisierung von Softwareport, hat eine Referenz zu einem TriggerPortTrigger

Konfiguration: Über einen TriggerPort werden Events einer Softwarekomponente von einer anderen aus gesetzt. Der Typ sagt aus, ob es ein Outputport oder Inputport ist.

Klasse ServerClientPort

Eigenschaften: enthält den Typ

Relationen: ist eine Spezialisierung von Softwareport

Konfiguration: Über einen ServerClientPort werden einer anderen Softwarekomponenten Funktionen zur Verfügung gestellt. Der Typ legt fest, ob der Port sendet oder empfängt.