

TargetLink  
 RTE\_APPL\_CODE (S12\_e1 \* (RTE\_Pim\_X(instance))

```
Controller/Controller_Runnable/(pcs) */
FUNC(void, RTE_APPL_CODE) controller_runnable(CONSTIP2CONST(Rte_CDS_Controller, RTE_CONST, RTE_CONST
instance)
{
```

```
/* SLocal: Default storage class for local variables | Width: 16 */
sint16 S12_e1 /* LSB: 2^-9 OFF: 0 MIN/MAX: -64 .. 63.998046875 */;
sint16 S12_si1 /* LSB: 2^-9 OFF: 0 MIN/MAX: -64 .. 63.998046875 */;
```

```
/* Sum: TL_Controller/Controller_Runnable/e1
# combined # TargetLink import: TL_Controller/Controller_Runnable/(ref)
# combined # TargetLink import: TL_Controller/Controller_Runnable/(pcs) */
S12_e1 = (sint16) (((sint32) Rte_IRead_PosController_RequiredSignals_ref(instance)) - ((sint32)
Rte_IrvRead_PosController_LinPos(instance)) >> 1);
LOG_VAR(1, _S12_e1, S12_e1);
```

```
/* Sum: TL_Controller/Controller_Runnable/si1 */
S12_si1 = (sint16) (S12_e1 + *(Rte_Pim_X(instance)));
```

```
/* Sum: TL_Controller/Controller_Runnable/sPI1
# combined # Gain: TL_Controller/Controller_Runnable/Ki1
# combined # Gain: TL_Controller/Controller_Runnable/Kp1 */
Rte_Pim_ACP_1(instance)->S12_sPI1 = (sint16) ((S12_e1 * ((sint16) Rte_CData_Kp(instance))) +
((sint16) (((sint32) S12_si1) * ((sint32) Rte_CData_Ki(instance)) >> 9)));
```

```
/* # combined # TargetLink outport: TL_Controller/Controller_Runnable/(upi) */
Rte_IWrite_PosController_ProvidedSignal_upi(instance, Rte_Pim_ACP_1(instance)->S12_sPI1);
```

```
/* Unit delay: TL_Controller/Controller_Runnable/xi1 */
*(Rte_Pim_X(instance)) = S12_si1;
```

```
/* Sum: TL_Controller
```

Neue Funktionen in TargetLink 3.5  
vereinfachen die Codegenerierung

# Zeichen setzen

Die neue Version 3.5 des dSPACE Seriercode-Generators TargetLink bietet als wichtigste Neuerungen die Unterstützung für Simulink-Aufzählungsdatentypen (Enum) zur transparenteren Modellierung, die AUTOSAR-4.1-konforme Entwicklung und die Generierung von mehrfach instanzierbaren Software-Komponenten. Darüber hinaus wird die Software-Einführung von TargetLink für einen großen Anwenderkreis spürbar vereinfacht und die Prozessdokumentation erleichtert.

#### Flexiblere Modellierung

Die neue TargetLink®-Version bietet substanzielle Erweiterungen für die Modellierung in Simulink®/Stateflow®. Eine wesentliche Neuerung ist insbesondere die Unterstützung von Simulink-Aufzählungsdatentypen (Enum), um hiermit „sprechendere“ Modelle zu erzielen (Abbildung 1). Der Anwender kann auf der Simulink- und Stateflow-Modellebene mit symbolischen Namen statt mit Zahlenliterals arbeiten, was die Lesbarkeit und Wartbarkeit der Modelle deutlich erhöht. Zudem stellt TargetLink 3.5 eine Komfort-Funktionalität zur Verfügung, um für beliebige, native Simulink-Blöcke auf eine sehr einfache Art C-Code zu hinterlegen, der dann bei der Seriercode-Generierung für den entsprechenden Simulink-Block verwendet wird. Der Anwender wird hierbei durch die Generierung von entsprechenden Dateifragmenten unterstützt, die er nur noch blockspezifisch ergänzen muss. Weitere Verbesserungen im Bereich der Modellierung sind beispielsweise die Unterstützung der Initialisierung von Stateflow-Charts, die deren Ausführung zum Zeitpunkt null festlegt, sowie die Zwischenspeicherung von Eingangssignalen („latched inputs“) für durch Funktionsaufrufe angesteuerte Subsysteme.

#### Praxisbewährte AUTOSAR-Unterstützung weiter ausgebaut

Auch mit TargetLink 3.5 werden alle in der Praxis relevanten AUTOSAR-Versionen unterstützt, wobei die AUTOSAR-Revision 4.1.1 neu hinzugekommen ist und die Unterstützung älterer Revisionen vollständig erhalten bleibt. Wie bei TargetLink üblich, können aus demselben Modell Software-Komponenten für unterschiedliche AUTOSAR-Versionen generiert werden, um die Wiederverwendung von Modellen in unterschiedlichen Projekten zu vereinfachen. Als grundlegend neues AUTOSAR-Feature bietet TargetLink 3.5 jetzt auch die Generierung von Software-Komponenten für die Mehrfachinstanziierung, (Abbildung 2). Hierbei wird der Komponenten-Code von TargetLink so erzeugt, dass mehrere Instanzen des Komponenten-Prototyps auf einem Steuergerät gebildet werden können, die alle denselben Code verwenden. Hierdurch und durch instanzspezifische AUTOSAR-Kalibrierparameter werden nicht nur Steuergeräte-Ressourcen gespart, sondern auch Testaufwände reduziert. Darüber hinaus ist die inkrementelle Code-Generierung für Software-Komponenten jetzt noch leistungsfähiger und flexibler ein-

setzbar. Ferner wurde das Zusammenspiel von TargetLink mit dSPACE SystemDesk® durch die Bereitstellung von TargetLink-spezifischen Validierungsregeln vereinfacht, so dass ein SystemDesk-Architekturmodell unmittelbar auf Konformität mit TargetLink überprüft werden kann.

#### Effizienterer Code und verbesserte Variantenunterstützung

Mit TargetLink 3.5 erhält der Anwender eine erweiterte Code-Optimierung für die Eliminierung von Zustandsvariablen mit statischer Speicherdauer, wodurch RAM-Speicherplatz eingespart wird. So können kostbare Steuergeräte-Ressourcen noch effizienter genutzt werden. Darüber hinaus weist TargetLink Verbesserungen in Hinblick auf die Konformität mit MISRA-Regeln auf, wobei alle im neuen MISRA-C:2012-Standard als verpflichtend („Mandatory“) gekennzeichneten Regeln eingehalten werden. Ein mit TargetLink 3.5 neu eingeführter Code-Generierungsreport hilft etwa in Bezug auf verwendete Code-Generierungsoptionen oder implizite Typ-Konvertierungen und Umskalierungen von Ports und deren Vorgängerblöcken. Letztere sind oft das Resultat einer Fehlspezifikation des Anwenders und können mit Hilfe



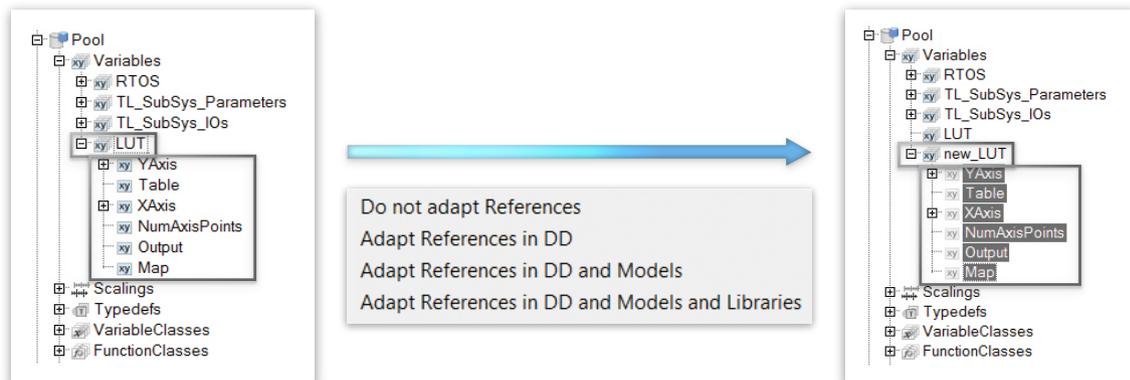


Abbildung 3: Vereinfachte Refaktorisierung (Strukturverbesserung) von Data-Dictionary-Spezifikationen wie die Anpassung von Objektnamen.

Mit TargetLink 3.5 wurde die praxisbewährte AUTOSAR-Unterstützung weiter ausgebaut. Ob AUTOSAR 4.1 oder Mehrfachinstanziierung von Komponenten – TargetLink bietet die Lösung.

Verbindungsline besteht oder der eigentliche Seriencode beeinflusst wird. Dies ist hilfreich, wenn fehlende Modell- und Code-Anteile zu Testzwecken durch „Stubs“ realisiert werden, die während der Simulation geeignet stimuliert werden müssen. Die Bedienbarkeit von TargetLink hat ebenfalls deutliche Verbesserungen erfahren, u.a. durch eine leistungsfähigere Auswahl von multiplen Data-Dictionary-Objekten zur Visualisierung und Bearbeitung sowie eine Refaktorisierungsfunktionalität (Strukturverbesserung) bei Änderung von Namen und Pfaden von Data-Dictionary-Objekten (Abbildung 3).

#### Vereinfachte Einführung von TargetLink 3.5 in Arbeitsgruppen

Mit TargetLink 3.5 fällt die Einführung von TargetLink für eine Vielzahl von Benutzern leichter, da klar zwischen TargetLink-Installationsdaten

und projektspezifischen Anpassungsdateien wie Hook-Funktionen getrennt wird. Die projektspezifischen Anteile können so sauber versioniert und zusammen mit der dann unveränderten TargetLink-3.5-Installation verwendet werden (Abbildung 4). Hierdurch können alle Projektmitarbeiter mit den gleichen Projekteinstellungen wie z.B. Hook-Funktionen arbeiten, wodurch sich die

Prozesssicherheit deutlich erhöht. TargetLink 3.5 unterstützt die zum Release-Zeitpunkt aktuellen vier MATLAB-Versionen, angefangen bei MathWorks R2013b bis hin zu MathWorks R2012a und ist wie üblich sowohl als 32-Bit- als auch 64-Bit-Version verfügbar. TargetLink 3.5 wird wie seine Vorgängerversionen ISO-26262- und IEC-61508-zertifiziert werden. ■

Abbildung 4: Das verteilte Arbeiten in großen Gruppen wird mit TargetLink 3.5 maßgeblich vereinfacht.

