

Restbussimulation im Prüfstandseinsatz

Eine neue Lösung war gefordert

Für den Automobilhersteller Volkswagen ergab sich 2010 die Notwendigkeit der Suche nach einer neuen Lösung für den Einsatz von Restbussimulationen (RBS) an den Prüfständen in ihrer Aggregatentwicklung. Die bestehende Lösung war bereits mehr als zehn Jahre im Einsatz und den aktuellen und zukünftigen Anforderungen nicht mehr gewachsen. Die neue Lösung sollte neben dem CAN-Bus auch LIN- und FlexRay-Bussysteme unterstützen. Zusätzlich waren noch Analog- und Digital-I/O, eine integrierte Toolkette sowie die Möglichkeit, eigene Usercodes und Funktionsblöcke integrieren zu können gefordert. Hierbei stand der Ansatz „Parametrieren statt Programmieren“ im Vordergrund. Dabei war das Ziel, den variablen Teil der Restbussimulation weitgehend durch Konfiguration und Parametrierung mit einem Standardtool erstellen und anpassen zu können.

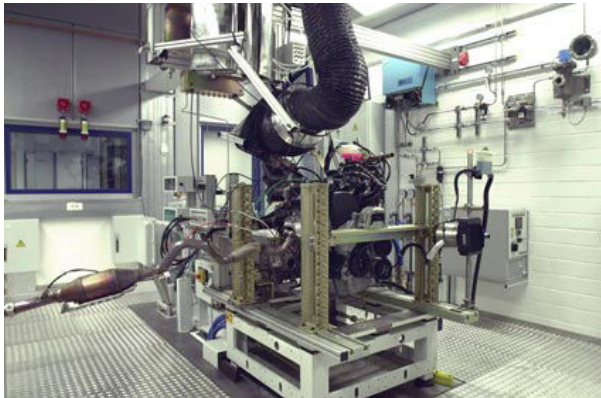


Bild 1: Aggregatprüfstand der Volkswagen AG

Die bestehende Restbussimulation war integraler Bestandteil des aktuellen Prüfstands-Automatisierungssystems (AuSy) und damit nur eingeschränkt nutzbar. Diese Einschränkungen sollten in einer neuen Lösung vermieden werden. Da alle Prüfstände über eine CAN-Bus-Architektur verfügen, war eine Gateway-Lösung erforderlich, welche die prüfstandsrelevanten Größen an das Automatisierungssystem weiterleitet bzw. die Signale der Restbussimulation entsprechend der Prüfstandsgrößen verändert.

In einem formalen Auswahlverfahren mit mehreren Mitwettbewerbern und deren aktuellen Marktprodukten, wurde nach Messung der jeweiligen Turnaround-Zeiten die autarke Box-Lösung mit einer Ethernet-Ankopplung an das Prüfstands-Automatisierungssystem bevorzugt. Entscheidend waren letztlich die Vorteile der Box-Variante, diese unabhängig vom AuSy zu betreiben und die Möglichkeit, die Box in der Nähe des Prüflings montieren zu

können, was den Verkabelungsaufwand sowie die notwendigen Kabellängen deutlich reduzierte.

Hardware- und Software aus einer Hand

Die Wahl fiel schließlich auf das basicCAN61plus-Modul aus der Serie61-Controller-Familie von GÖPEL electronic. In der gewählten Konfiguration bietet diese Box drei CAN-Schnittstellen (2 x Motor-CAN, 1 x AuSy-CAN), eine LIN-Schnittstelle (Nebenaggregate) und zwei FlexRay-Schnittstellen sowie je sechs analoge Ein- und Ausgänge und je acht digitale Ein- und Ausgänge. Darüber hinaus verfügt die Box über eine leistungsstarke 600 MHz PowerPC CPU für die Ausführung der Restbussimulation sowie über eine Gigabit-Ethernet-Schnittstelle und eine USB-2.0-Schnittstelle. Ein „Klemme-15“-Relais wurde ebenfalls in der Box integriert.



Bild 2: RBS-Box basicCAN61plus

Für die Erstellung der Restbussimulation nutzt Volkswagen die Net2Run-Software von GÖPEL electronic. Diese ermöglicht es in wenigen Minuten, eine lauffähige statische Restbussimulation zu erstellen. Hierzu werden die notwendigen Informationen aus den entsprechenden DBC-, LDF- oder FIBEX-Bordnetzdatenbanken importiert und die zu simulierenden Steuergeräte und deren Botschaften ausgewählt. Checksummenberechnungen und Botschaftszähler, sowie die notwendigen LIN- und FlexRay-Sende-Schedules werden hierbei automatisch konfiguriert. Ebenso werden PDU-Multiplexer, das gesamte PDU-Triggering sowie das FlexRay-Update-Bit Handling automatisch konfiguriert. Ist dies erfolgt, müssen nur noch die Signale ausgewählt werden, welche zur Laufzeit im RBS-Programm zugreifbar sein sollen. Danach wird eine so genannte RBS-Datei generiert. Diese RBS-Datei kann auf der Box gespeichert werden und wird anschließend vom RBS-Programm geladen und ausgeführt. Alternativ kann die RBS auch direkt aus der Net2Run-Software heraus gestartet werden.

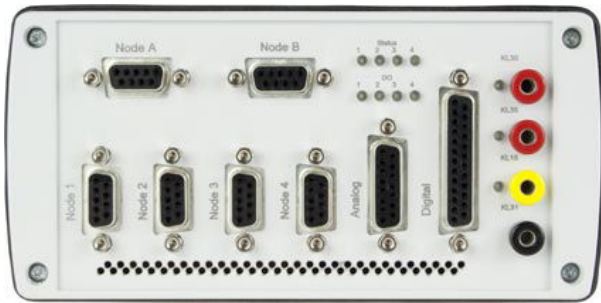


Bild 3: Frontansicht basicCAN61plus

Restbussimulation durch modulare Lösung

Um die Restbussimulation über die AuSy-CAN-Schnittstelle steuern zu können, muss die RBS-Box von GÖPEL electronic als Gateway konfiguriert werden. Dies geschieht über den Gateway-Editor in Net2Run. Hierzu werden die zu übertragenden Signale in einer Routing-Tabelle gemappt und ggf. mit einen zusätzlich Skalierungsfaktor oder Offset versehen. Auch diese Information wird in der RBS-Datei gespeichert.

Um die RBS-Box unabhängig vom PC oder AuSy betreiben zu können, wird auf der Karte ein sogenanntes RBS-Programm (Usercode) benötigt. Dies ist ein kleines C oder C++ Programm, welches die GÖPEL-API-Funktionen nutzt, um auf die Schnittstellen der GÖPEL-electronic-Hardware und deren Firmware-Funktionen zuzugreifen. Volkswagen verwendet RBS-Programme mit umschaltbaren Varianten für z.B. Handschalter oder Automatikgetriebe. Ein einfaches RBS-Programm initialisiert alle Schnittstellen und aktiviert bzw. deaktiviert im Bedarfsfall die Terminierungswiderstände. Anschließend wird die RBS-Datei geladen und die RBS gestartet. Sobald gültige Botschaften vom Motorsteuergerät erkannt wurden, startet das RBS-Programm die Tx-PDU Gruppen der Restbussimulation bzw. stoppt diese wieder, sobald der Motor keine Informationen mehr sendet. Darüber hinaus kann ein RBS-Programm die analogen und digitalen I/O steuern und auch signalbasierte Modelle oder Funktionen für die dynamische Restbussimulation enthalten. Solche Funktionen können auch in Bibliotheken zusammengefasst werden, sodass sie in verschiedenen RBS-Programmen wiederverwendet werden können.

Als Werkzeug für die Erstellung von On-Board-RBS-Programmen setzt Volkswagen die Net2Run-IDE von GÖPEL electronic ein. Diese auf Eclipse basierende Entwicklungsumgebung enthält alle notwendigen Bausteine zum Erstellen und Debuggen von C und C++ Programmen für die RBS-Box. Die Programme werden auf der RBS-Box unter dem QNX-Echtzeit-Betriebssystem ausgeführt. Alle für die Restbussimulation erforderlichen Funktionen sind in der GÖPEL API enthalten und können genauso unter Windows verwendet werden. Zusätzlich stehen Nutzern neben den Standard-C-Bibliotheken auch zahlreiche, zumeist POSIX-kompatible QNX-Bibliotheken zur Verfügung.

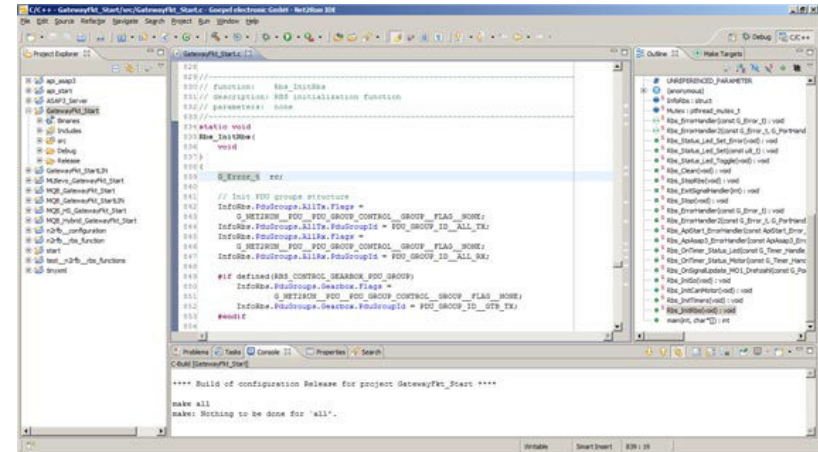


Bild 4: Net2Run-IDE (Code-Perspektive)

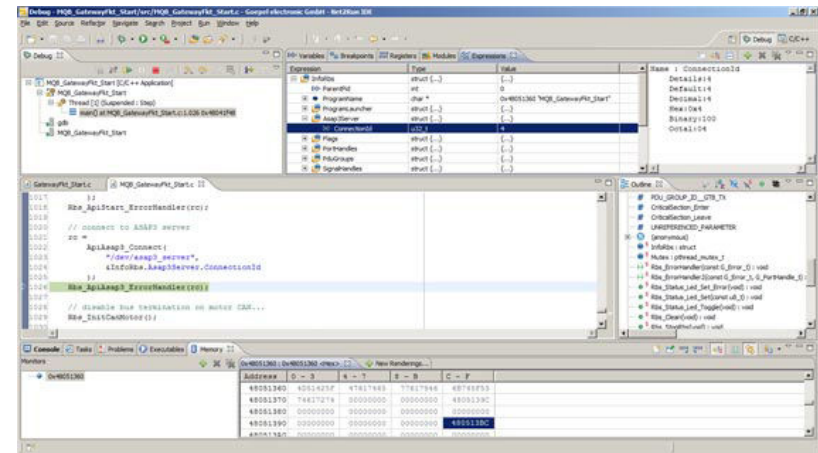


Bild 5: Net2Run-IDE (Debug-Perspektive)

Die RBS-Programme sind so aufgebaut, dass sie mit unterschiedlichen RBS-Dateien funktionieren. Jedoch gibt es für unterschiedliche Aggregate und Bussysteme entsprechend angepasste RBS-Programme. So wird z.B. zwischen CAN- und FlexRay-Steuergeräten sowie Triebsträngen mit LIN-Sub-Bus für Nebenaggregate unterschieden, aber auch funktional zwischen Otto-, Diesel- und Hybrid-Antrieben. Hieraus ergeben sich zahlreiche Kombinationsmöglichkeiten aus RBS-Programmen und RBS-Dateien. Diese werden als RBS-Projekte zusammengefasst und in einer XML-Datei definiert. In dieser Datei wird auch festgelegt, welches RBS-Projekt beim Starten der RBS-Box automatisch geladen werden soll. Auf der RBS-Box befindet sich hierzu ein kleines Programm – der so genannte „Program-

Launcher“. Dieser wird beim Booten der RBS-Box ausgeführt und sorgt dafür, dass das ausgewählte RBS-Programm gestartet wird und diesem die zugehörige RBS-Datei übergeben wird. Darüber hinaus lässt sich der Program-Launcher fernsteuern. Zum einen ist es möglich, die Restbussimulation über den CAN-Bus zu wechseln, zum anderen lässt sich dies mit dem GÖPEL-electronic-Tool „RBS-Selector“ vom PC aus bewerkstelligen. Aktuelle Status- und Fehlermeldungen werden von den RBS-Programmen über einen FIFO-Puffer an den PC gesendet und werden mittels des On-Board-Monitor-Tools angezeigt.

ID	RBS-Project	RBS-Name	Start RBS...	RBS-Application File Name	RBS-Simulation File Name
1	VW_ANT_496_HS	Handschalter		./bin/GatewayFkt_Start	./tbs/VW_ANT496_AG_HS_Gateway.rbs
2	VW_ANT_496_AG	Automatikgetriebe		./bin/GatewayFkt_Start	./tbs/VW_ANT496_AG_HS_Gateway.rbs
3	VW_ANT_496_PP	Powepack		./bin/GatewayFkt_StartLIN	./tbs/03_ANT496_PP_Gateway.rbs
4	VW_ANT_MQB_HS	Handschalter		./bin/MQB_GatewayFkt_Start	./tbs/VW_ANTMQB_HS_AG_Gateway.rbs
5	VW_ANT_MQB_AG	Automatikgetriebe		./bin/MQB_GatewayFkt_Start	./tbs/VW_ANTMQB_HS_AG_Gateway.rbs
6	VW_ANT_MLB_HS	Handschalter		./bin/MLB_GatewayFkt_Start	./tbs/VW_ANTMLB_HS_AG_Gateway.rbs
7	VW_ANT_MLB_AG	Automatikgetriebe		./bin/MLB_GatewayFkt_Start	./tbs/VW_ANTMLB_HS_AG_Gateway.rbs
8	VW_ANT_MQB_PP	Powepack		./bin/MQB_GatewayFkt_StartLIN	./tbs/VW_ANTMQB_PP_Gateway.rbs
9	VW_ANT_496_PP_HY_V...	Powepack_mt_Hybrid		./bin/GatewayFkt_StartLIN	./tbs/VW_ANT_496_PP_HY_VW361_Gateway.rbs
10	VW_MQB_AG_HYB_V7_10F	MQB_mt_Hybrid		./bin/MQB_Hyb_Start	./tbs/VW_MQB_AG_HYB_V7_10F.rbs
11	FUB_TEST_VW_ANTMQB...	FUB Test		./bin/MQB_HS_GatewayFkt_Start	./tbs/VW_ANTMQB_HS_AG_Gateway_neu.rbs
12	FUB_TEST_VW_MQB_AG	FUB Test		./bin/MQB_Hybrid_GatewayFkt_Start	./tbs/VW_MQB_AG_HYB_V7_10F.rbs
20	Aud_MLB_Evo	MLB Evo mit FlexRay		./bin/MLB_Evo_FlexRay_Start	./tbs/Aud_MLB_evo_FlexRay.rbs

Bild 6: RBS-Selector

Hausinternes Automatisierungssystem zur Prüfablaufsteuerung

Zur Steuerung des Prüfablaufs sowie zur Erfassung von Messwerten wird das hauseigene Volkswagen-Prüfstands-Automatisierungssystem VENUS eingesetzt. Die Kommunikation erfolgt über das CAN-Protokoll. Das Automatisierungssystem sendet über den CAN-Bus alle notwendigen Parameter und Führungsgrößen, welche über das Gateway der RBS-Box den dynamischen Teil der Restbussimulation beeinflussen. So lassen sich auch LIN-gesteuerte Nebenaggregate und FlexRay-Steuergeräte über CAN fernsteuern. Die Messwerte der einzelnen Busse werden über das Gateway an VENUS gesendet und dort ausgewertet. Neben den hauseigenen Prüfständen nutzt Volkswagen auch Prüfstände von externen Dienstleistern, welche über andere Prüfstands-Automatisierungssysteme verfügen; auch hier lässt sich meist mit geringem Aufwand eine CAN-Anbindung realisieren.

Alternativ zur CAN-Kommunikation hat Volkswagen zusammen mit der Firma GÖPEL electronic einen ASAP3-Server für die RBS-Box entwickelt. Dieser läuft als Dienst auf der RBS-Box und bietet einen einfachen Zugriff auf die Signale der Restbussimulation. ASAP3 ist ein ASAM-spezifiziertes Protokoll zur Kalibrierung und Messwerterfassung von Steuergerätesoftware. Der ASAP3-Standard wird von zahlreichen Kalibrierungswerkzeugen und Prüfstands-Automatisierungssystemen namhafter Anbieter sowie der hausinternen VENUS-Software unterstützt. Der ASAP3-Client verbindet sich über die Gigabit-Ethernet-

Verbindung mit dem ASAP3-Server und kann anschließend auf die Tx-Signale der Restbussimulation als Parameter zugreifen – die Rx-Signale werden als Messwerte behandelt. Welche Parameter bzw. Messwerte in der RBS verfügbar sind, erfährt der ASAP3-Client aus der A2L-Datei (ebenfalls ein ASAM-Standard). Net2Run bietet hierzu einen integrierten A2L-Export für die Versionen 1.51 und 1.60 des Standards, welcher alle in der Signalliste enthaltene Signale als Parameter (Tx) bzw. Messgröße (Rx) exportiert. Für den Signalgang verwendet der ASAP3-Server die Befehle der GÖPEL API. Die Konfiguration erfolgt über eine XML-Konfigurationsdatei, die mittels des TinyXml-Parsers eingelesen wird. Über die API des Program-Launchers ist es zudem möglich, mittels des ASAP3-Servers die laufende Restbussimulation zu wechseln. Zur Ethernet-Kommunikation wurde die POSIX-kompatible Socket-I/O-Bibliothek des QNX-Betriebssystems verwendet.

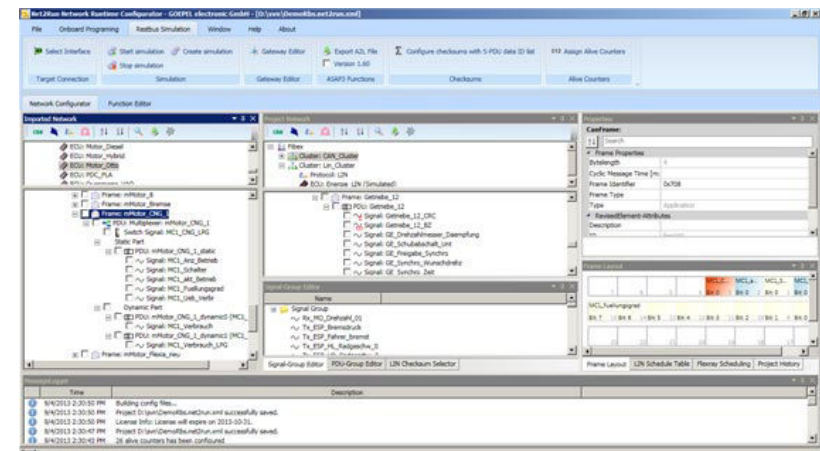


Bild 7: Net2Run-Konfigurator

Abschlussbemerkung

Im Rahmen der Integrationsphase wurden mit relativ geringem Aufwand Restbussimulationslösungen geschaffen, welche sich an verschiedene Prüfstandsanforderungen anpassen lassen. Hierbei wurde bewusst auf proprietäre Lösungen verzichtet und stattdessen auf offene Standards, Standardtools und Standardschnittstellen gesetzt. Das Ergebnis ist ein flexibler Baukasten von Komponenten, der sich beliebig erweitern und an neue Anforderungen anpassen lässt.

Der Autor:



Thomas Sedlaczek studierte Informatik an der Fachhochschule in Wolfsburg und Electronic Design an der University of Glamorgan in Süd Wales. Er sammelte einige Jahre Erfahrung als Softwareentwickler bei Sanken Power Systems Ltd. in Cardiff und als FAE bei dSPACE Ltd. in Cambridge. Seit Januar 2008 ist er als Softwareentwickler bei der der GÖPEL electronic GmbH in Jena für die FlexRay Lösungen und Net2Run verantwortlich.

Thomas Sedlaczek

Dipl.-Ing.(FH), B.Eng.(Hons)