

Mit UPStudio erfolgreich Multiprozessor-Systeme entwickeln

Mit den heutigen Single-Prozessor-Software-Entwicklungssystemen ergeben sich zahlreiche Schwachstellen und Fehlerquellen, welche den Entwicklungsprozess ganz wesentlich erschweren. Deshalb gibt es jetzt das neue Multiprozessor-Entwicklungssystem Universal Programming Studio von PMS (Prahm Microcomputer Systeme). Es bietet eine besonders einfache und effiziente Entwicklung von vernetzten Computersteuerungen, damit Produkte schneller und sicherer die Marktreife erreichen können.

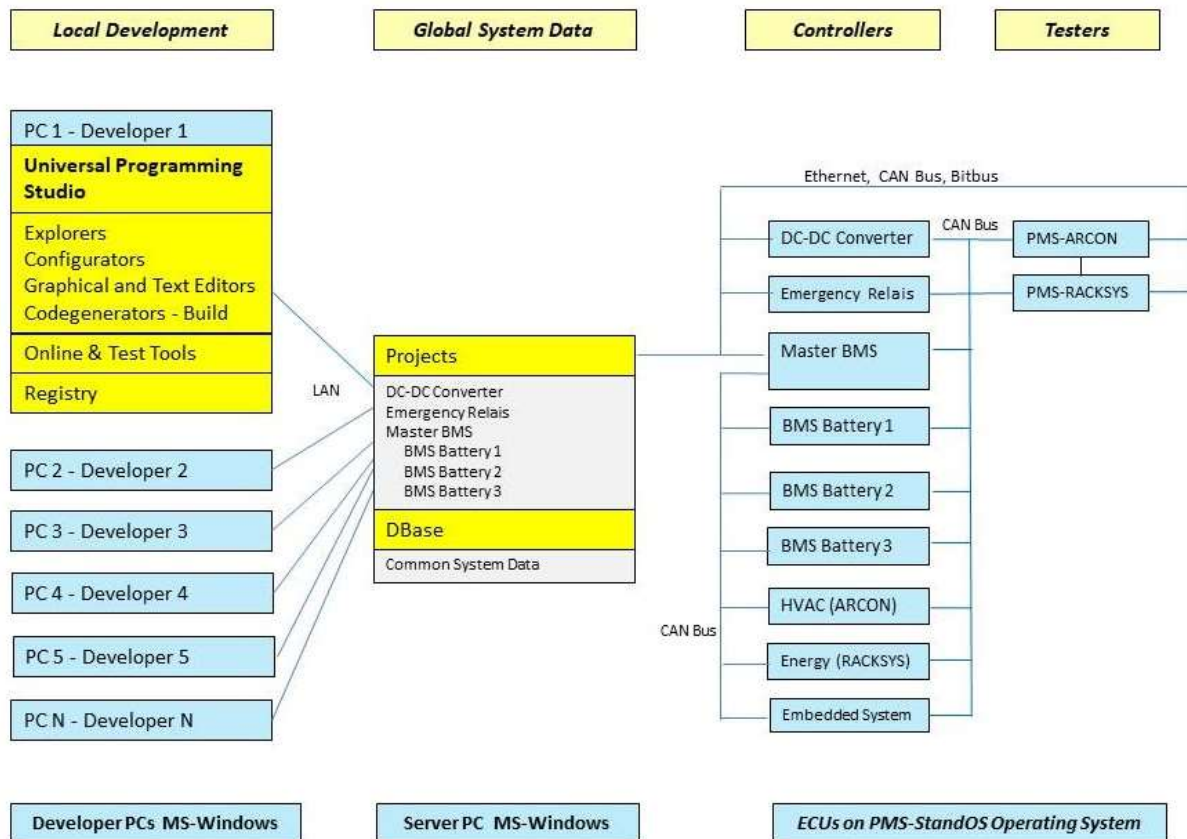


Abbildung 1 - Struktur des Multiprozessor-Entwicklungssystems Universal Programming Studio

Bessere Archivierung und Versionisierung mit dem neuen Universal Programming Studio

Die folgende kurze Beschreibung vom Universal Programming Studio (kurz UPStudio) soll die Struktur des Multiprozessor-Entwicklungssystems verdeutlichen.

Wie in der Abbildung 1 gezeigt, befindet sich auf jeden Entwickler-PC eine registrierte Edit-Version vom Universal Programming Studio. Im „bin“- Verzeichnis des Entwickler-PCs befinden sich dabei alle *.exe- Dateien inklusive der UPStudio IDE. Und im „Help“- Verzeichnis befinden sich alle Hilfedateien vom UPStudio.

Damit mit den Konfiguratoren und Grafik-Editoren Zielcode erzeugt werden kann (C, C++, C# und Header-Dateien) wird nach Bedarf ein Generation-Release mit bedienerfreundlichen Tools eingerichtet. Für ein Online- Debugging und –Monitoring wird entsprechend ein Online- Release interaktiv eingerichtet.

Mit UPStudio erfolgreich Multiprozessor-Systeme entwickeln

Auf dem Server PC befinden sich dann zentral alle Daten des Systems. Die Baumstruktur der Projekt-Verzeichnisse auf dem Server PC entspricht der hierarchischen Hardware-Struktur der Bus-Netzwerke des Systems.

Multi-Prozessor kontra Single-Prozessor-Entwicklungssysteme

Eine ernsthafte Fehlerquelle bei Single-Prozessor-Entwicklungssystemen ist zum Beispiel das Aufkommen von mehrfachen IDs bei der USD Diagnose (UDS = Unified Diagnostic Services). Das Durcheinander bei den IDs entsteht beispielsweise wegen dem Zeitverzug, bis neue IDs im Gesamt-UDS-Katalog aufgenommen werden. So können schwer erkennbare Fehler entstehen, welche dann gravierende Folgen haben können. Solche Fehler werden oft erst entdeckt, wenn nach einem Unfall die ECUs des Drive-Chains ausgelesen werden (ECU = Electronic Control Unit).

Diese Fehlerquelle hat mit dem PMS System Configurator ein Ende. Denn dieses Tool überwacht alle Eigenschaften der Bus-Systeme und zeigt alle Fehler an. Fehler wie ungültige Service IDs, mehrfache Daten IDs, abweichende Baud-Raten sowie andere fehlerhafte Bus-Einstellungen. Diese Fehler sind dann durch interaktive Eingaben schnell behoben.

Eine zweite Fehlerquelle ist das Matchen von parallelen Entwicklungspfaden bei den heutigen Archivierungs- und Versionisierungssystemen. Auch diese Fehlerquelle ist mit dem UPStudio jetzt kein Problem mehr, siehe die folgende Beschreibung.

Archivierungs- und Versionisierung jetzt einfacher und sicherer gemacht


Für die geordnete Embedded Software-Entwicklung werden heute noch Archivierungs- und Versionisierungs-Tools eingesetzt. Wenn ein Entwickler die Aufgabe hat, eine Applikation zu erstellen, macht er mit diesen Tools in der Projekt-Datenbank einen Check-Out von den Software-Modulen in welchen die Veränderungen geschehen sollen. Solange er daran arbeitet, sind diese Dateien für alle anderen Kollegen gesperrt. Aber. Sehr oft müssen schon parallel andere Applikationen in den gleichen Dateien erstellt werden, wesshalb man einen parallelen Entwicklungszweig einrichtet. Am Ende der Entwicklungen müssen die Dateien wieder zusammen gematched werden. Und das ist erfahrungsgemäß eine ernsthafte Fehlerquelle.

Die Lösung für dieses Problem ist im UPStudio die Trennung der PCs von der Projektdatenbank. Die Entwicklung der Applikationsentwicklung geschieht auf den PCs der Entwickler, während die gesamte Software auf dem Server-PC verbleibt. Jedes Projekt ist im Grunde eine Standard-Core, wie dieser von BMW bekannt ist. Nur mit dem Unterschied, dass die Projekte dort in Unterverzeichnissen angeordnet sind – so wie die Struktur der kaskadierten Netzwerke es vorgibt.

Da die Entwicklung direkt in der Projektdatenbank geschieht, kann man sich das Ein- und Aus-Checken ersparen und damit auch die Archivierungs- und Versionisierungs-Tools. Denn die Aufgaben der Archivierungs- und Versionisierungs-Tools übernimmt nun das Universal Programming Studio.

Mit UPStudio erfolgreich Multiprozessor-Systeme entwickeln

Aufgabenverteilung mit der Developers-Datenbank



	Developer Items	Developer Data
	Company Name	Renergy
	Department	Development
	Developer	Herbert Winter
	Email	hwinter@renergy.com
	Phone	0172 824864
	Active	Still active
	Start time	27.10.2023 18:27:39
	Exit time	
	Project	C:\UPStudio\Projects\ArconRack\StringBMS\CAN3\HVAC
	Password	*****
	Access control	0x00002488
*		

Abbildung 2 - Developers-Datenbank für alle Entwickler des Projekts

Beim Start einer neuen Produkt-Version trägt der Projektleiter alle Entwickler in der Developers-Datenbank ein, beziehungsweise überarbeitet die vorherigen Einstellungen, siehe Abbildung 2. Der Projektleiter kann die alte Version archivieren, um so eine Projekthistorie für spätere Analysen zu erhalten. Mit der Projekt- und Access-Control-Eintragung legt er fest, auf welches Projekt und in welcher Datei der Entwickler Schreib- und Leserechte erhält. Im Access Control Eintrag sind bis zu 32 Bit-Einträge möglich, wobei jedes Bit festlegt, welche Rechte der einzelne Entwickler bekommt. Für besonders kritische Zugriffe kann der Projektleiter noch ein Zugriffspasswort einrichten.

Mit UPStudio erfolgreich Multiprozessor-Systeme entwickeln

Transparente Projektüberwachung mit der Developers Presence Liste

Universal Programming Studio	System Configurator	StandOS Configurator	Diagnose Configurator				
Company Data							
Company Name	Dudolf Systems	Dudolf Systems	Dudolf Systems	Dudolf Systems	Reenergy	Reenergy	
Department	Development	AppDev	Development	Development	Development	Development	
Developer	Klaus Jensen	Frank Hansen	Gert Hilversum	Markus Peters	Herbert Winter	Fred Johnson	
Email	KJensen@Dudolf.de	fhansen@dudolf.de	gh@dudolf.de	mpeters@dudolf.de	hw@reenergy.com	fj@reenergy.com	
Phone	040 822456	0171 263268	040 82 34 56	040 824532	012 659821	012 659825	
Active	Inactive	Still active	Still active	Still active	Still active	Still active	
Start time	23.10.2023 14:25:57	24.10.2023 17:40:31	24.10.2023 17:4...	31.10.2023 14:5...	27.10.2023 18:2...	27.10.2023 17:17:28	
Exit time	24.10.2023 17:21:31						
Project	C:\UPStudio\Projects...	C:\UPStudio\Projects...	C:\UPStudio\Proj...	C:\UPStudio\Proj...	C:\UPStudio\Proj...	C:\UPStudio\Projects...	
*							

Abbildung 3 - Die Developers Presence Liste erübrigt das Ein- und Aus-Checken von Dateien

Statt den Check-In und Check-Out wie in den alten Archivierungs- und Versionisierungssystemen gibt es im Universal Programming Studio die Developers Presence Liste für alle UPStudio Tools, siehe Abbildung 3. In der Anwesenheitsliste der UPStudio IDE (Integrated Development Environment) kann man in der Active-Zeile sehen, welcher Entwickler im Moment mit UPStudio arbeitet. Wenn zum Beispiel ein Entwickler mit dem System Configurator die Netzwerkstruktur bearbeitet, und ein weiterer Entwickler ebenfalls auf sein PC den System Configurator aufrufen will, wird der Aufruf des Tools verhindert. Zur Konfliktlösung bekommt der hinzugekommene Entwickler ein Info-Fenster zu sehen, in dem ihm mitgeteilt wird, welcher Entwickler dieses Tool bereits nutzt. Mit allen Kommunikationsdaten des ersten Entwicklers wird der neue Entwickler aufgefordert sich mit dem Kollegen auszutauschen, um zu erfahren, wann er das Tool nutzen kann und um eventuell weitere Absprachen zu machen.

Bei der Software-Entwicklung mit den drei grafischen Editoren des UPStudios und dem integrierten Texteditor wird genauso verfahren, wie im Beispiel mit dem System Configurator beschrieben. Denn selbstverständlich können auch im UPStudio nicht zwei Entwickler das gleiche grafische oder textuale Programm zur gleichen Zeit bearbeiten.

Konflikte werden von vornherein vermieden, indem standardmäßig alle grafischen und textualen Quellcode-Module einen Eingangs- Funktionsaufruf mit fest definierten Ein- und Ausgängen hat. Denn dies erleichtert ganz wesentlich die Zuordnung von Applikationen zu den verschiedenen Entwicklern. Diese Funktionsaufruf-Methode hat sich bereits sehr lange und sehr gut in der Automation bewährt und ist auch bereits ein weltweiter Industrie- Standard. Die Firma Prahm Microcomputer Systeme stellt bereits mit seinem Firmenlogo in Form eines Funktionsbausteins seine auf Funktionalität basierende Firmenpolitik dar.

Mit UPStudio erfolgreich Multiprozessor-Systeme entwickeln

Whitebox-Test, ECU-Test und System-Test

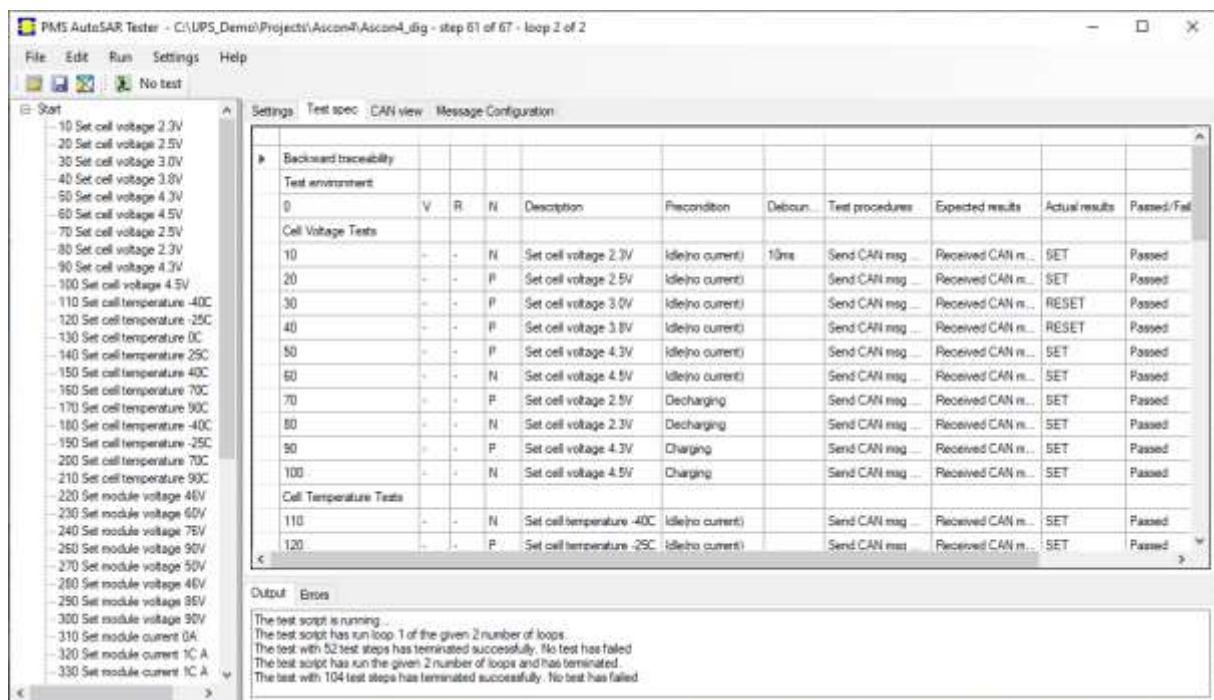


Abbildung 4 - Automatisierte Test mit dem AutoSAR Tester- Testskript und Testdokumentation alles in einer Tabelle

Der AutoSAR Tester kann seine Testskripte aus MS-Excel oder Doors importieren, um dann die von den Skript- Tabellen gesteuerten Tests automatisiert ablaufen zu lassen, siehe Abbildung 4. Dabei kann der Tester beliebige Schleifendurchläufe der Tests vorgeben, einen Teststopp zu einem bestimmten Zeitpunkt oder endlos bis zu einen manuellen Abbruch angeben. Alle Passed/Failed- Testergebnisse werden zusammen mit allen Umgebungsdaten direkt in der Tabelle eintragen. Diese Testspezifikation kann wieder zurück nach MS-Excel oder Doors exportiert werden.

Tests können auch direkt in den Grafik- Editoren debugged und getestet werden. Wenn zum Beispiel der Entwickler mit der IDE des UPStudio ein Funktionsbaustein aufgerufen hat, wird dieser PMS-FB2-Funktionsbausteineditor aufgerufen und im Funktionsbaustein-Editor wird das aufgerufene Funktionsbaustein-Programm dargestellt. Im rechten Teil des Grafikeditors werden in den Eigenschaftsfenstern alle Inputs und Outputs des Funktionsbausteins angezeigt. Der Anwender kann dort für verschiedene Whitebox-Tests alle Inputs mit passenden Daten belegen. Dann kann der Anwender sich durch das Funktionsbausteinprogramm steuern (mit Step-Over oder Step-Into) oder einen kompletten Zyklusschritt machen. Im Letzteren Fall sieht der Entwickler im Eigenschaftsfenster direkt die veränderten Outputs abhängig von den vor jedem Zyklusschritt gesetzten Input-Daten. Selbstverständlich steht auch ein Watchfenster für die Sichtung von internen Daten zur Verfügung. Für diesen Funktionsbaustein-Test kann der Anwender für seine Whitebox-Test ein lokales Build auf seinem PC machen. Diese Testmöglichkeiten gibt es auch für die anderen Grafikedatoren und für Tests von textuellen Programmen geschrieben in C, C++, C# und IEC-61131-AWL.

Mit UPStudio erfolgreich Multiprozessor-Systeme entwickeln

Die zweite Test-Methode ist der ECU-Test, welcher normalerweise nach einen der regelmässigen Archivierungs- und Build- Zyklen gemacht wird. Dazu wird auf dem Server-PC ein Build für das im Explorer eingestellte Projekt durchgeführt. Eventuelle Konflikte mit anderen Entwicklern werden durch die eingebaute Intelligenz von UPStudio vermieden. Der Entwickler kann dann die zum Projekt gehörende ECU direkt vor Ort an seinem PC testen.

Und bei der dritten Test- Methode, dem System-Test, kann der Entwickler im Testlabor ein Systemtest mit allen vernetzten ECUs durchführen. Für den finalen Test im PKW, werden über die Diagnoseschnittstelle des Autos alle Stell- und Testdaten während des automatisierten Tests übertragen und auf dem portablen PC gespeichert und angezeigt.

Der periodische Sicherungs- und Integrationslauf

Für den periodischen Integrationslauf kann der Projektleiter und der Integrator eine UPStudio- Applikation aufrufen, mit welcher zu erkennen ist, ob einer der Entwickler noch an einen der UPStudio Tools arbeitet. Falls dies der Fall ist, kann der Integrator sich der Reihe nach durch alle noch offenen und ungetesteten Projekte durchsteppen. Er muss dann die entsprechenden Entwickler gesondert ansprechen.

Der Integrator kann mit allen Projekten, welche den Status „Ready for Integration“ haben, einen automatisierten Sicherungslauf machen und mit diesen Projekten den Build für den Endkunden vornehmen.

PMS Prahm Microcomputer Systeme

www.prahm-ms.de

info@prahm-ms.de