

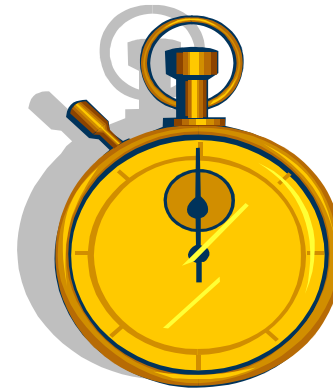
Neue Wege der Testautomatisierung

MC-ST: Framework für Systemtest-Automatisierung mit UML und LabVIEW

Kosten

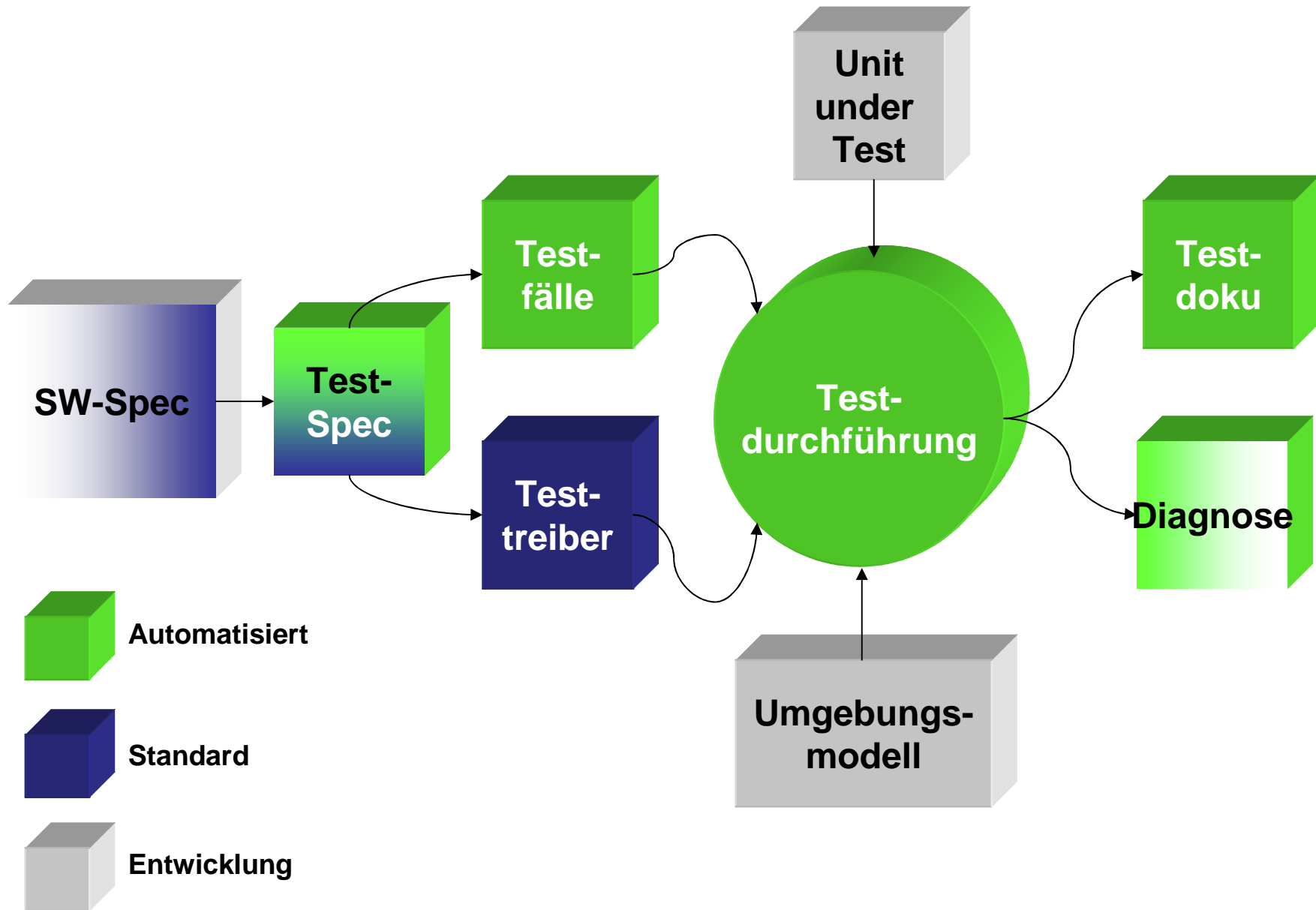


Zeit



Qualität

Q



Automatisierte Systemtests mit MicroConsult Systemtest, MC-ST

MC-ST ist ein Framework für automatisierte Systemtests mit UML und LabVIEW.

Es besteht aus dem Test Management Center, Test Executor und Test Configurator von MicroConsult.

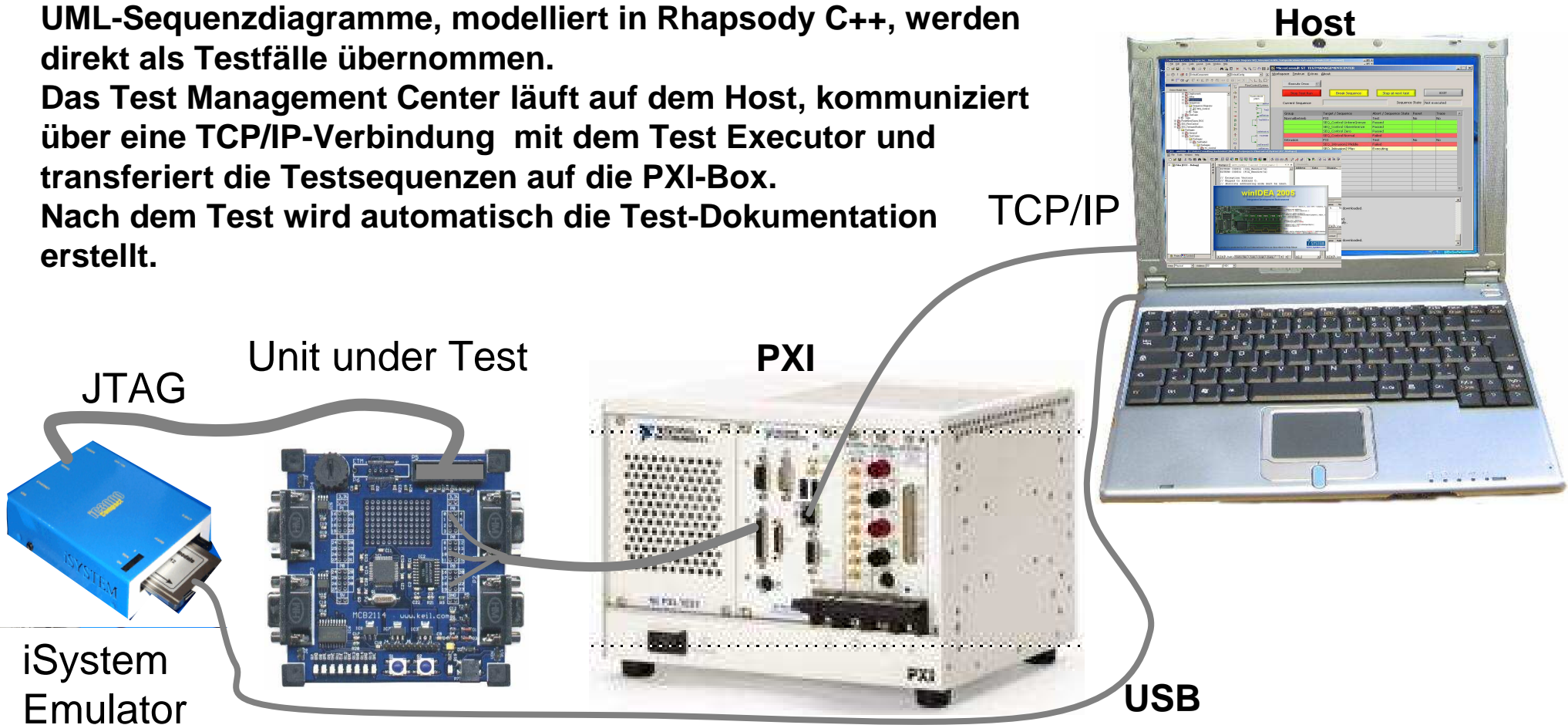
Der Test Executor läuft auf der PXI-Box. Er beinhaltet die Testtreiber und führt die Testsequenzen, modelliert in Rhapsody C++, aus. Er simuliert die Umgebung.

Mit dem Test Configurator können die Testfälle parametrisiert und zu Testsequenzen zusammengestellt werden.

UML-Sequenzdiagramme, modelliert in Rhapsody C++, werden direkt als Testfälle übernommen.

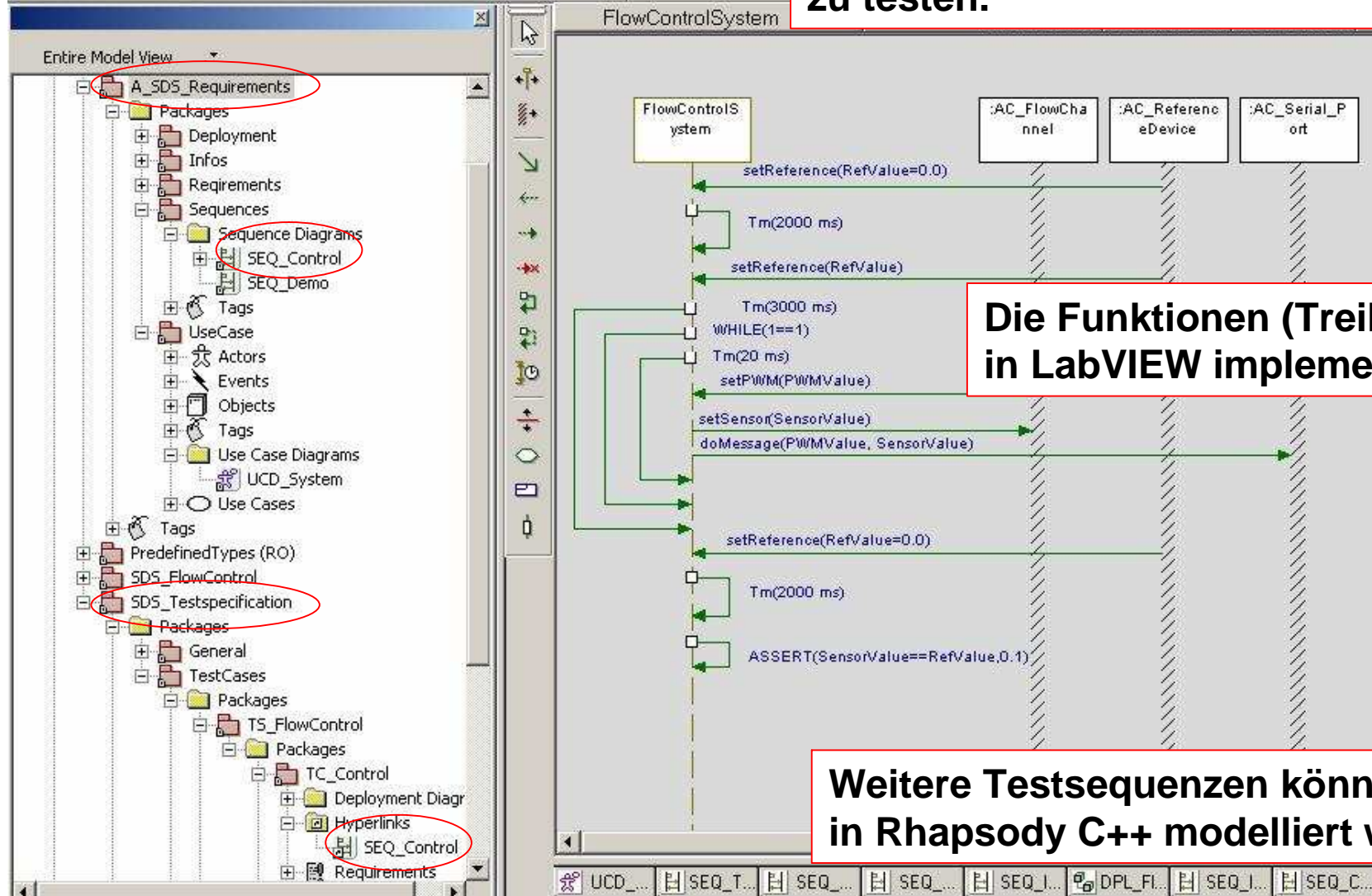
Das Test Management Center läuft auf dem Host, kommuniziert über eine TCP/IP-Verbindung mit dem Test Executor und transferiert die Testsequenzen auf die PXI-Box.

Nach dem Test wird automatisch die Test-Dokumentation erstellt.



Rhapsody in C++ by I-Logix Inc. - FlowControl.rpy - [Sequence Diagram: SEQ_Control in A_SDS_Requirements::Sequences]

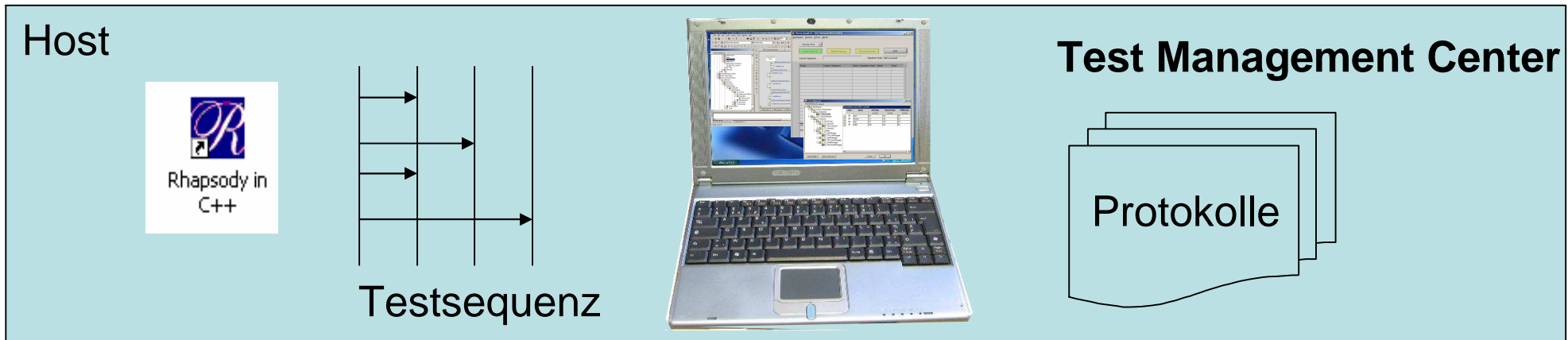
Sequenzen aus der Anforderungsanalyse werden als Testfälle übernommen, um die geforderte Funktionalität des Systems zu testen.



Die Funktionen (Treiber) müssen in LabVIEW implementiert werden

Weitere Testsequenzen können in Rhapsody C++ modelliert werden.

Softwarespezifikation -> Testspezifikation -> Testfälle



Rhapsody in C++ by I-Logix Inc. - [Sequence Diagram: SEQ_Control in A_SDS_Requirements::Sequences]

MC-ST TestConfigurator

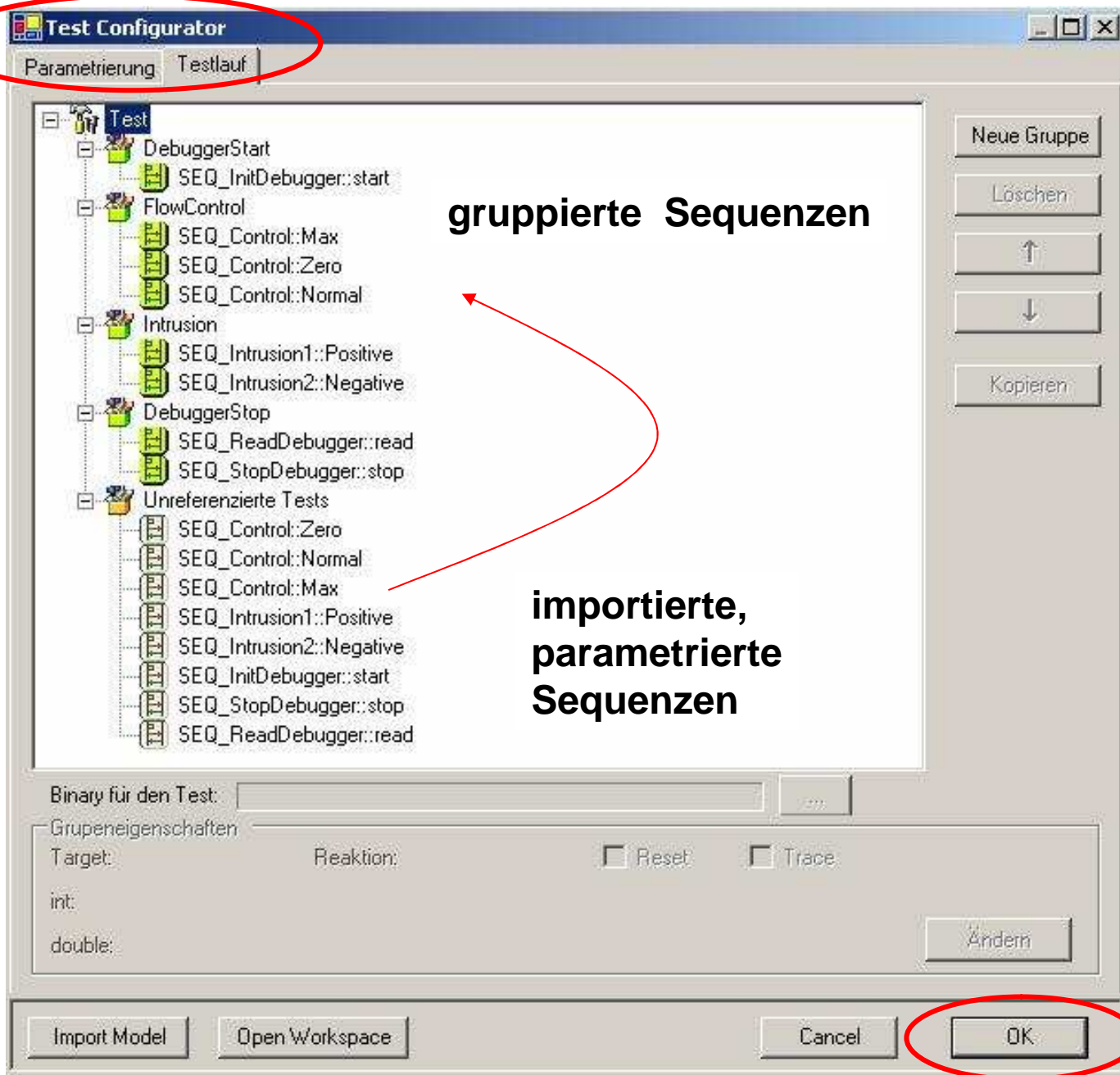
Requirements

Link

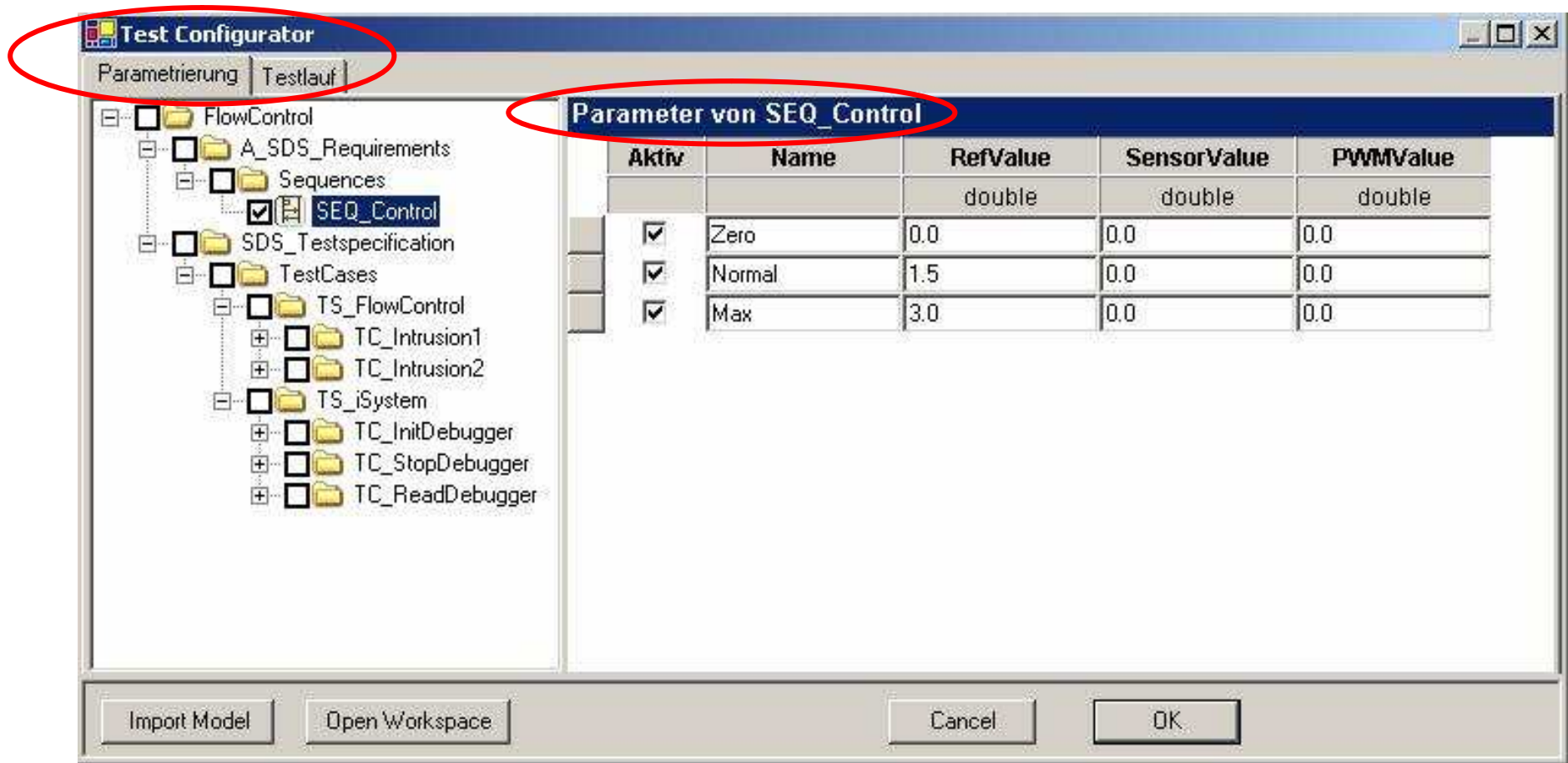
Test Spec

Sequenzen aus Rhapsody C++ werden in den Test Configurator des MC-ST importiert.

Testsequenzen werden von Rhapsody C++ in den Test Configurator importiert



Die Testsequenzen können zu Gruppen zusammengefasst und in eine beliebige Reihenfolge gebracht werden.



Ein Sequenzdiagramm kann für mehrere Testfälle mit verschiedenen Parametern benutzt werden.

Test Management Center

MicroConsult ST TESTMANAGEMENTCENTER

workspace Testrun Extras About

Execute Once

Start Test Run

Break Sequence

Stop at next test

EXIT

Current Sequence

Sequence State Not executed

Group	Target / Sequence	Abort / Sequence State	Reset	Trace
FlowControl	PXI	Test	No	No
	SEQ_Control Max	Not executed		
	SEQ_Control Zero	Not executed		
Intrusion	PXI	Test	No	No
	SEQ_Intrusion1 Positive	Not executed		
	SEQ_Intrusion2 Negative	Not executed		

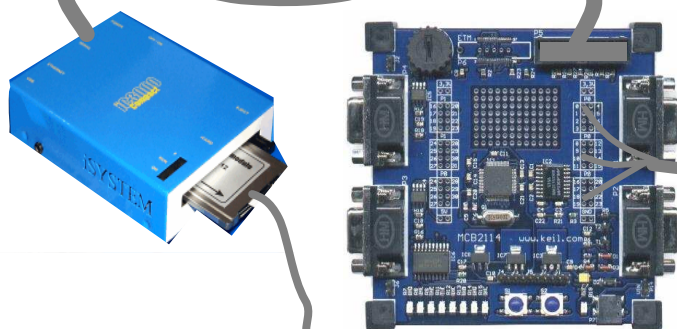
Die Testsequenzen werden in das Test Management Center geladen.

Host

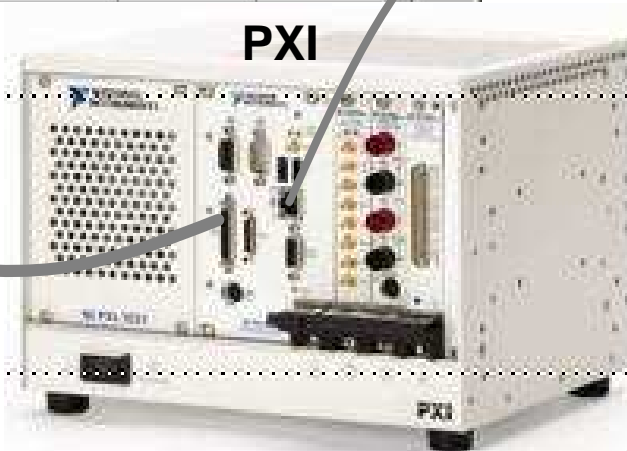


TCP/IP

JTAG



PXI



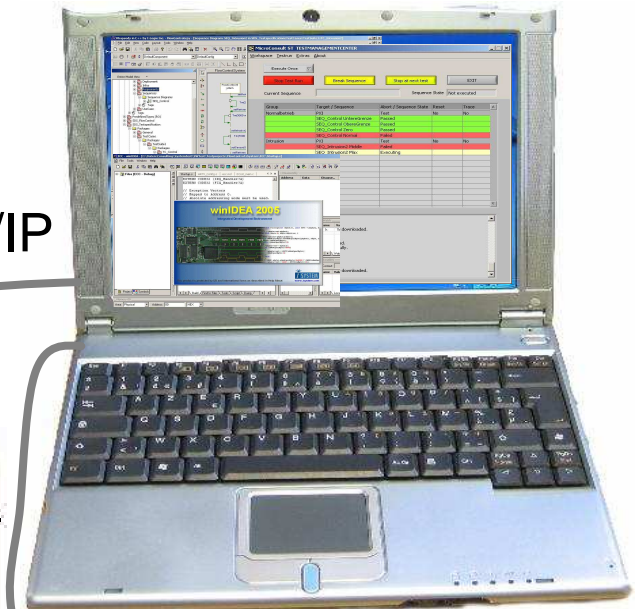
USB



Der Test Executor und die Treiber werden auf die PXI-Box geladen.

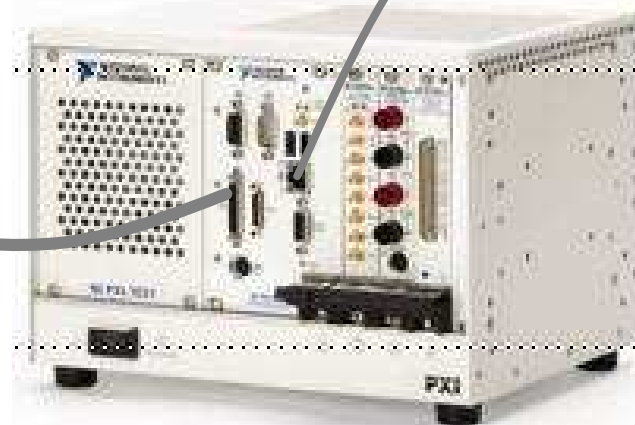
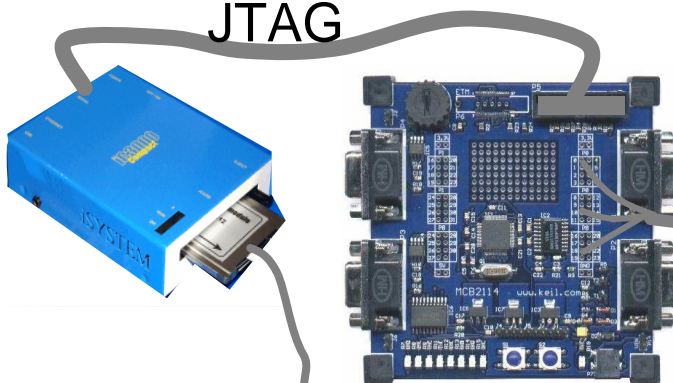
Das Test Management Center überträgt die Test an den Test Executor auf die PXI-Box. Die Tests werden ausgeführt.

TCP/IP

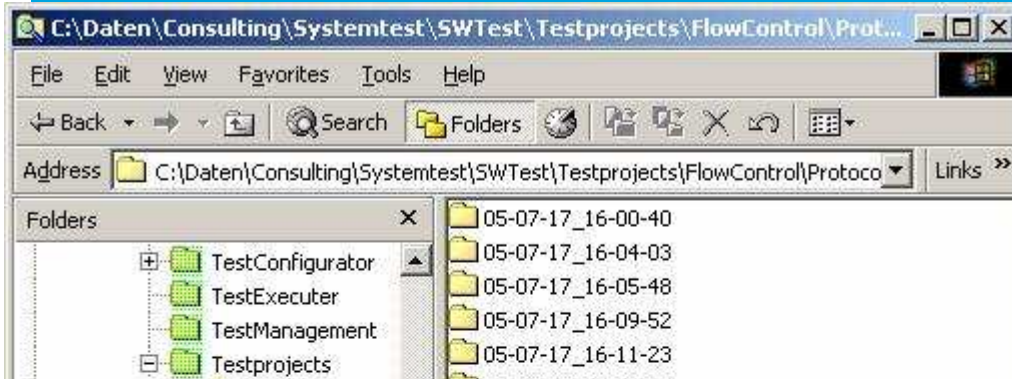


USB

JTAG



Nach dem Testlauf werden die automatisch erstellten Protokolle in einem Protokoll-Directory gespeichert.



```

=====Sequence State=====
SEQUENCE: SEQ_Intrusion2 Positiv
STATE: Passed
MESSAGE:
  executed till 15
TRACE:
09:32:30.922 IT      0  Flow Control started.
09:32:31.509 IT      1  A_SDS_Requirements::UseCase::FlowControlSystem.FlowControlSystem.setReference
09:32:31.509 IT      1  No return,0.000000
09:32:31.512 FL      2  wait-Initial 2000, ID 1
09:32:31.512 FL      2
09:32:31.513 IT      3  Timer wait/RST, ID 1
09:32:33.493 IT      3
09:32:33.494 IT      4  A_SDS_Requirements::UseCase::FlowControlSystem.FlowControlSystem.setReference
09:32:33.494 IT      4  No return,1.500000
09:32:33.495 IT      5  wait-Initial 3000, ID 2
09:32:33.495 IT      5
09:32:33.496 IT      6  Timer wait/RST, ID 2
09:32:36.467 IT      6
09:32:36.468 FL      7  A_SDS_Requirements::UseCase::AC_FlowChannel.AC_FlowChannel.setIntrusion2
09:32:36.468 FL      7  No return,0.700000
09:32:36.469 IT      8  wait-Initial 7000, ID 3
09:32:36.469 IT      8
09:32:36.470 IT      9  Timer wait/RST, ID 3
09:32:43.407 IT      9
09:32:43.407 FL     10  A_SDS_Requirements::UseCase::AC_FlowChannel.AC_FlowChannel.setsensor
09:32:43.424 IT     10  No return,1.499108
09:32:43.427 FL     11  A_SDS_Requirements::UseCase::FlowControlSystem.FlowControlSystem.setReference
09:32:43.427 FL     11  No return,0.000000
09:32:43.427 IT     12  wait-Initial 2000, ID 4
09:32:43.427 IT     12
  
```

Das Traceprotokoll zeigt jeden Schritt des Testlaufes.

Testprotokoll		lfd. Nr.:	Default Workspace Name_86
		Datum:	17.07.2005 17:23:13
Projekt:			
Projekt-Nr.:	Flow Control 001		
Projektname:	Flow Control Test Project		
Beschreibung:			
Test project for flow control. Test driver with data management.			
vorgelegt von			
Name/Abt.:	Rosenthal	Telefon:	+49(0)89 450617 62
		E-Mail:	A.rosenthal@microconsult.de
Ausgeführte Testfälle:			
Beschreibung der Testfälle	- SEQ_Control: Standard Use Case for controller - Countinous test		
Testfälle Statistik:	Klassifikation:	Absolut	% bezogen auf Anzahl
	Anzahl:	244	100
	Ausgeführt:	243	100
	Passed	243	100
	Failed	0	0
	Execution Error	1	0
Ausführung:			
Testobjekt:	Flow Channel Control		
Testumgebung: (Konfiguration, Test- umgebung, Hardware)	HIL, PXI, PXI-6221 MDAQ		

```

Debugger.log - Notepad
File Edit Format Help

[[variables]
k=0
[Registers]
R13=5F676B
[PWM_val_array]
PWM_val_array(0)=0
PWM_val_array(1)=0
PWM_val_array(2)=0
PWM_val_array(3)=0
PWM_val_array(4)=0
PWM_val_array(5)=0
PWM_val_array(6)=0
PWM_val_array(7)=0

```

Das Debugger Logfile enthält vom iSystem-Emulator gelesene Register- und Variablen-Werte.

Ein Testprotokoll nach IEEE 829 wird automatisch erstellt.

SEQ_Control.tst - Notepad

```

[sequence]
Name=SEQ_Control Norma

[parameters_int]
2000 const
20 const
1 const
3000 const

[parameters_double]
1.5 RefValue
0 sensorvalue
0 PWMValue
0.0 const
0.1 const

[parameters_string]

[steps]
1 1 2 2 0 40000004
7 0 3 4 1 80000001 3 3
8 0 4 1 1
1 1 5 2 0 40000001
7 0 6 4 3 80000004 3 d
4 0 7 4 80000003 0 80000003 d
7 0 8 4 2 80000002 3 b
1 2 9 2 0 40000003
1 3 a 2 0 40000002
1 4 b 3 0 40000003 40000002
8 0 c 1 2
0 0 6 0
8 0 e 1 3
1 1 f 2 0 40000004
7 0 10 4 4 80000001 3 10
8 0 11 1 4
6 0 12 4 40000002 0 40000001 400
                    
```

Test Configurator

Parametrierung | Testlauf

FlowControl

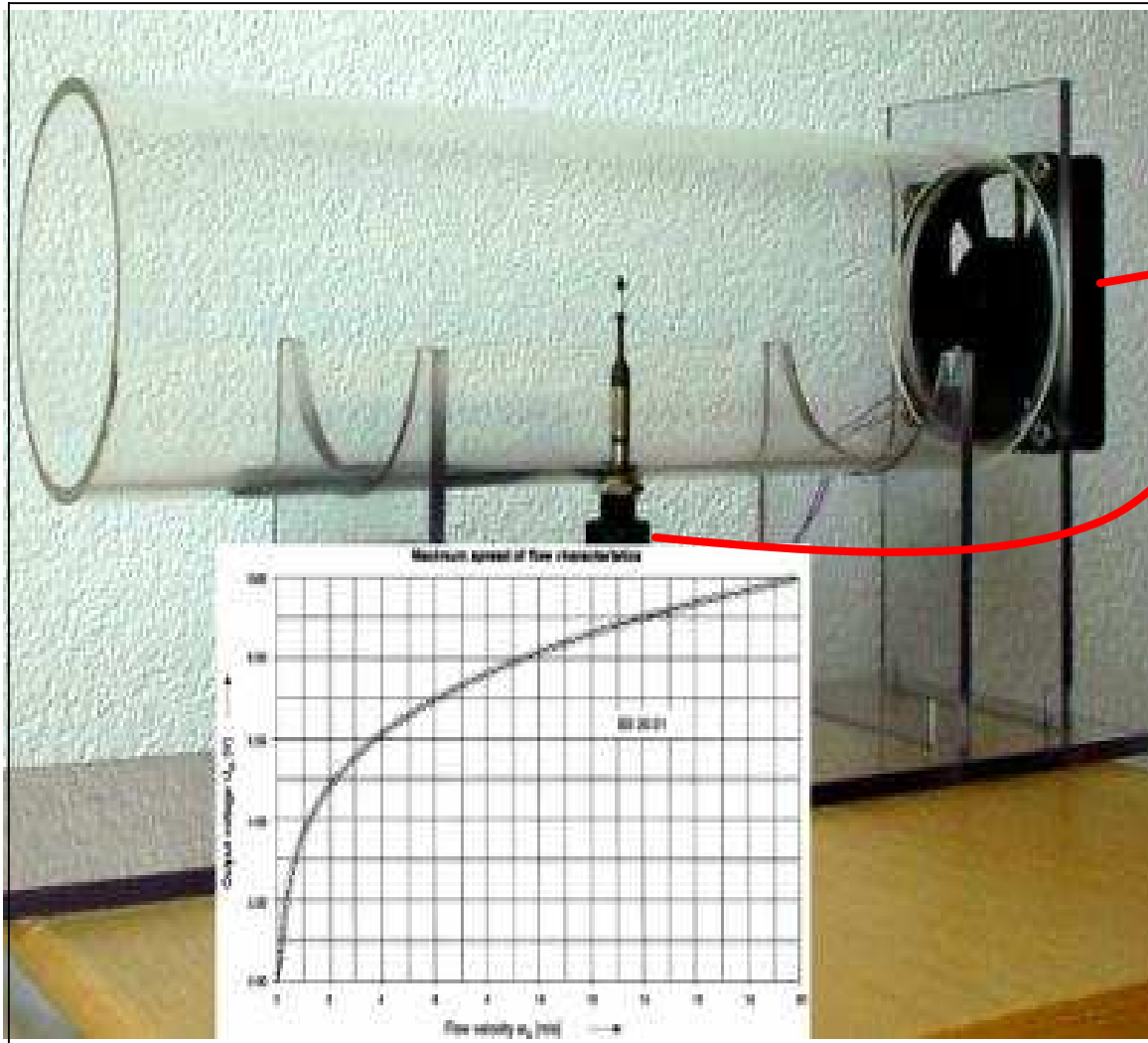
- A_SDS_Requirements
 - Sequences
 - SEQ_Control
- SDS_Testspecification
 - TestCases
 - TS_FlowControl
 - TC_Intrusion1
 - TC_Intrusion2
 - TS_System
 - TC_InitDebugger
 - TC_StopDebugger
 - TC_ReadDebugger

Parameter von SEQ_Control

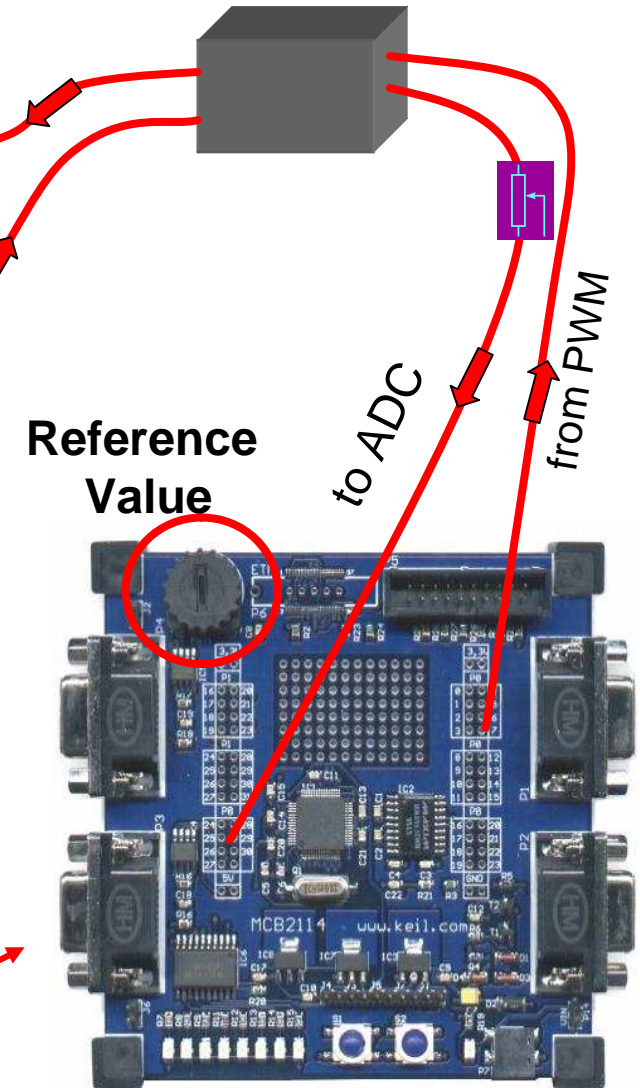
Aktiv	Name	RefValue	SensorValue	PWMValue
		double	double	double
<input checked="" type="checkbox"/>	Zero	0.0	0.0	0.0
<input checked="" type="checkbox"/>	Normal			
<input checked="" type="checkbox"/>	Max			

Testtreiber setReference(RefValue)

Konstanter Luftstrom im Strömungskanal



Control Device



Konstanter Luftstrom im Strömungskanal

PXI



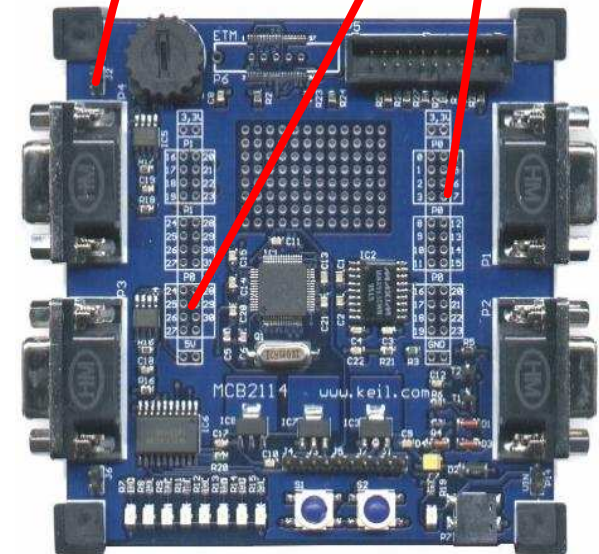
Umgebungsmodell

Control Device

Reference Value

to ADC

from PWM



Konstanter Luftstrom im Strömungskanal

PXI



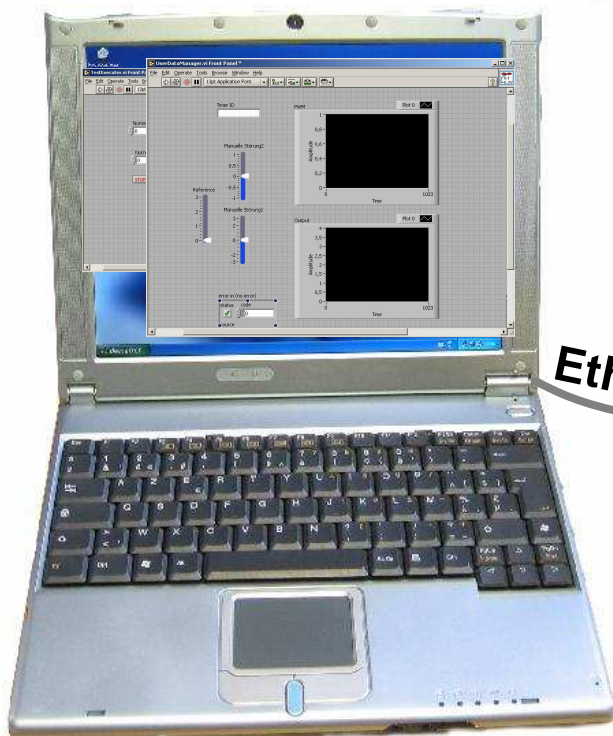
Umgebungsmodell

PXI



Control Device

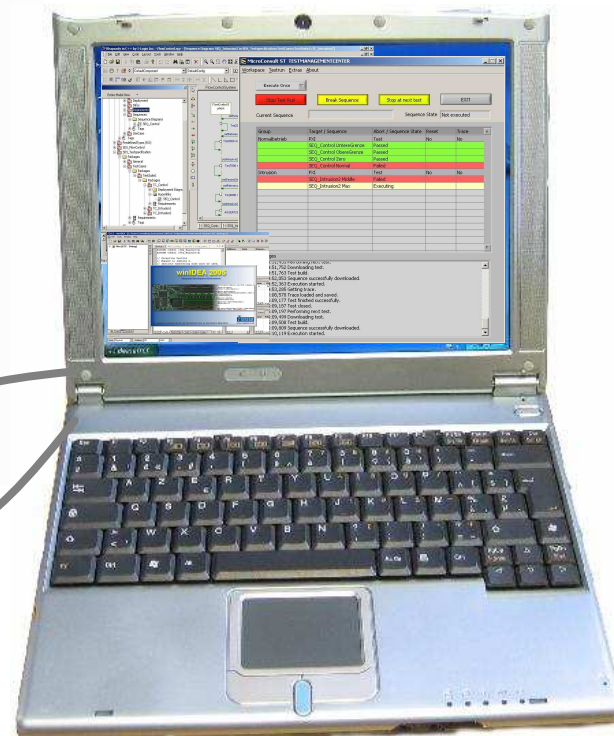
Visualisierung



PXI

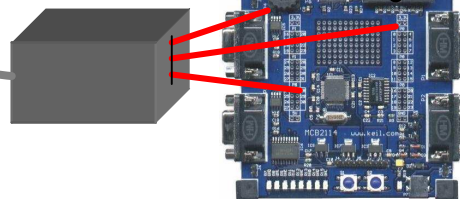
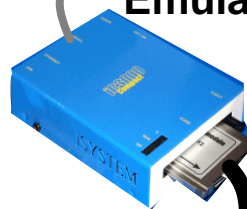


Host



Ethernet

iSystem Emulator

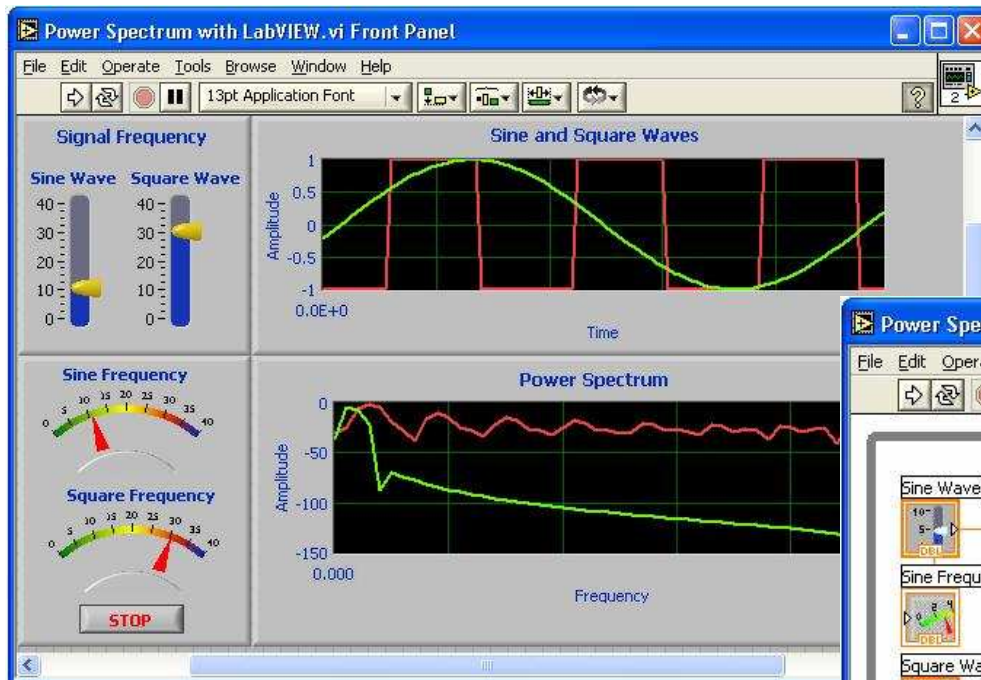


Rhapsody in C++
Test Management Center
iSystem winIDEA

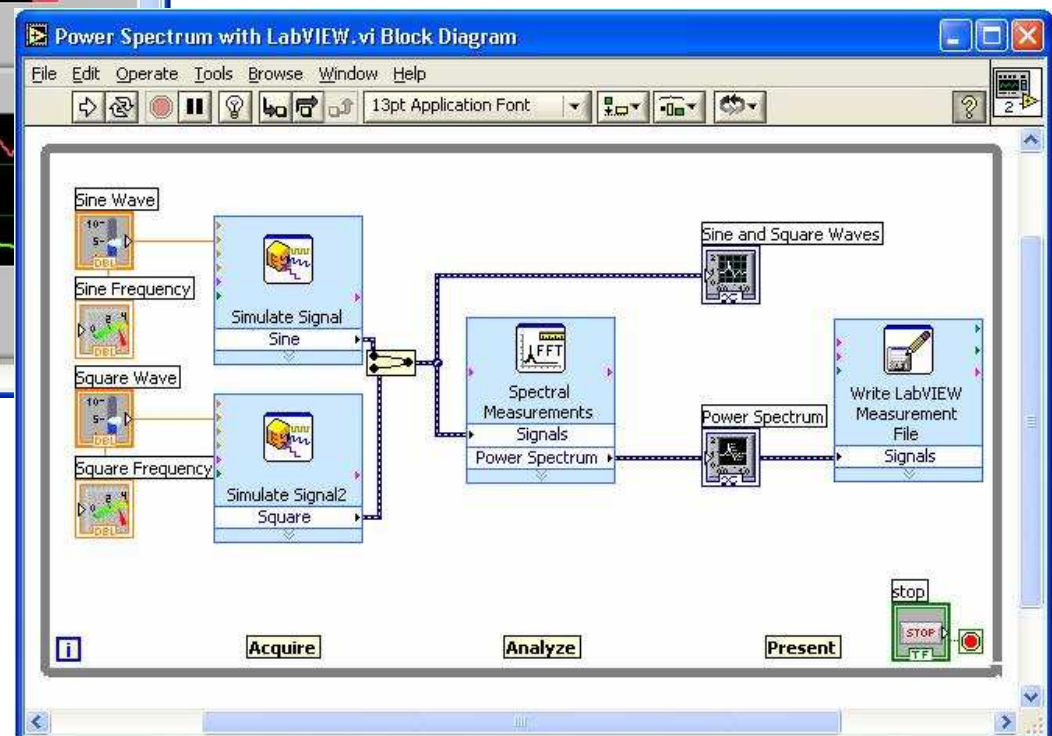
PXI (PCI eXtensions for Instrumentation) ist eine offene, PC-basierte Plattform für Messtechnik und Automatisierungssysteme mit mehr als 1000 I/O-Cards von verschiedenen Herstellern.



Mit dieser grafischen Umgebung erstellen Entwickler flexible, skalierbare Anwendungen für Test, Messung und Kontrolle.



User Interface

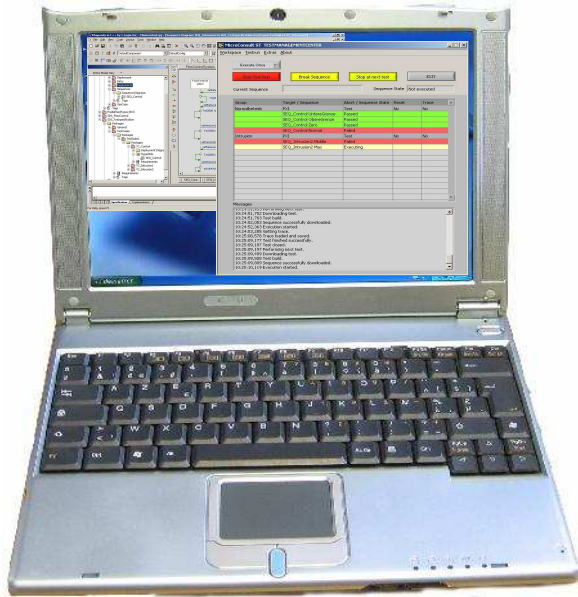




- Download
- Flash programming
- Program start/stop
- Variable read/write
- Register read/write
- Profiling

LabVIEW controls Debugger





PC-Sequenzer als SIL

- keine Echtzeit

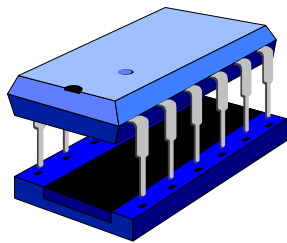
PXI-System als HIL

- ≥ 1 ms



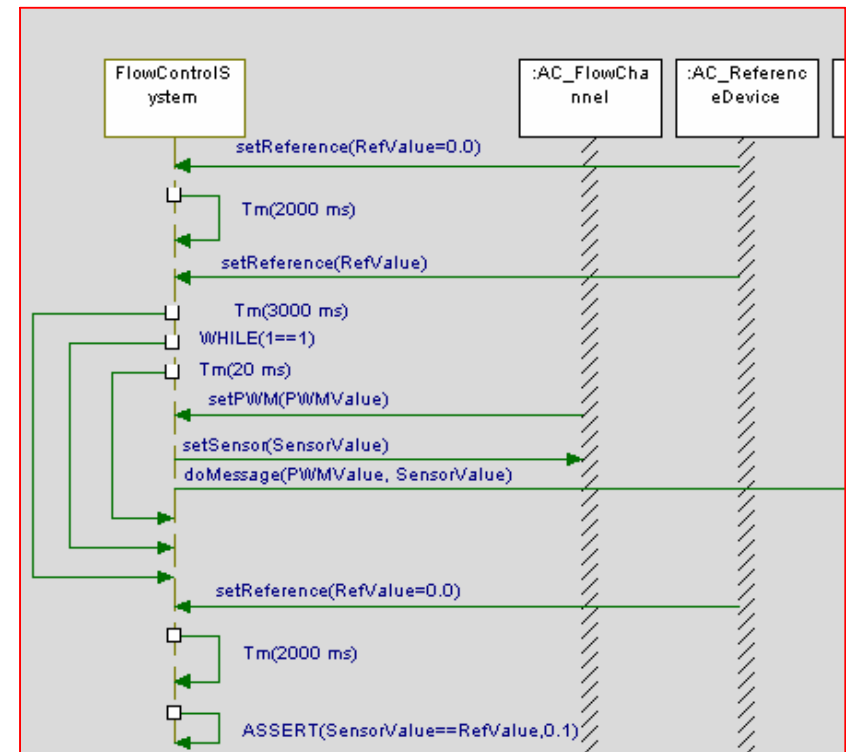
FPGA als HIL

- ≥ 25 ns



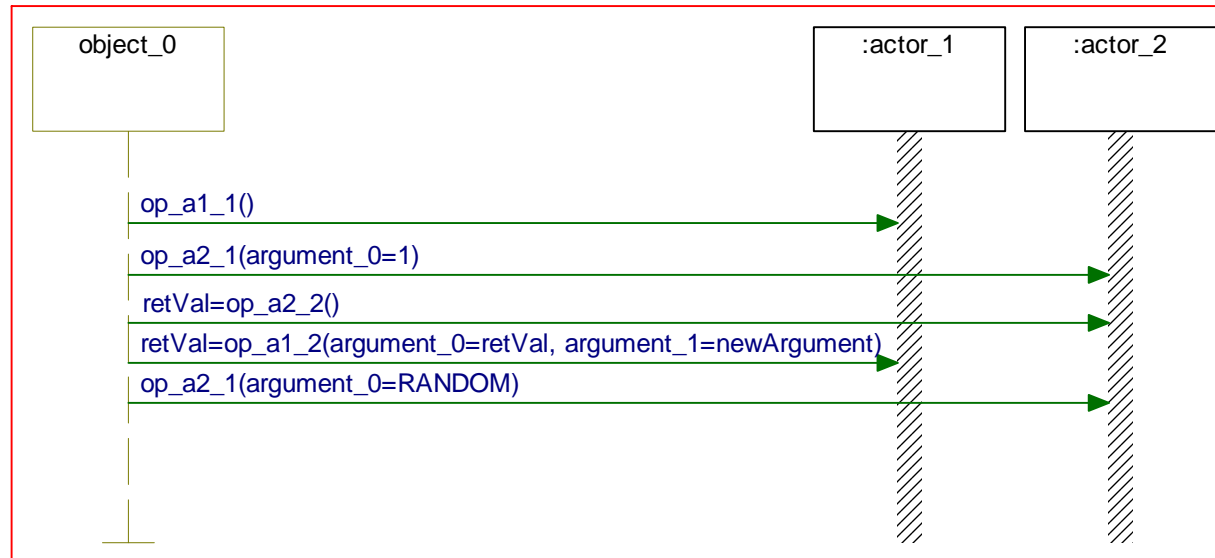
Sequenzdiagramme können folgende Operationen beinhalten, die das MC-ST beim Test ausführen kann:

- Aufruf von Methoden (mit Ein- und Rückgaben)
- For-Schleifen
- WHILE-Schleifen
- If-Anweisungen
- Assert
- Wait
- TimeOut (Watchdog)



Zulässig sind folgende Bedingungen:

<, >, <=, >=, !=, ==



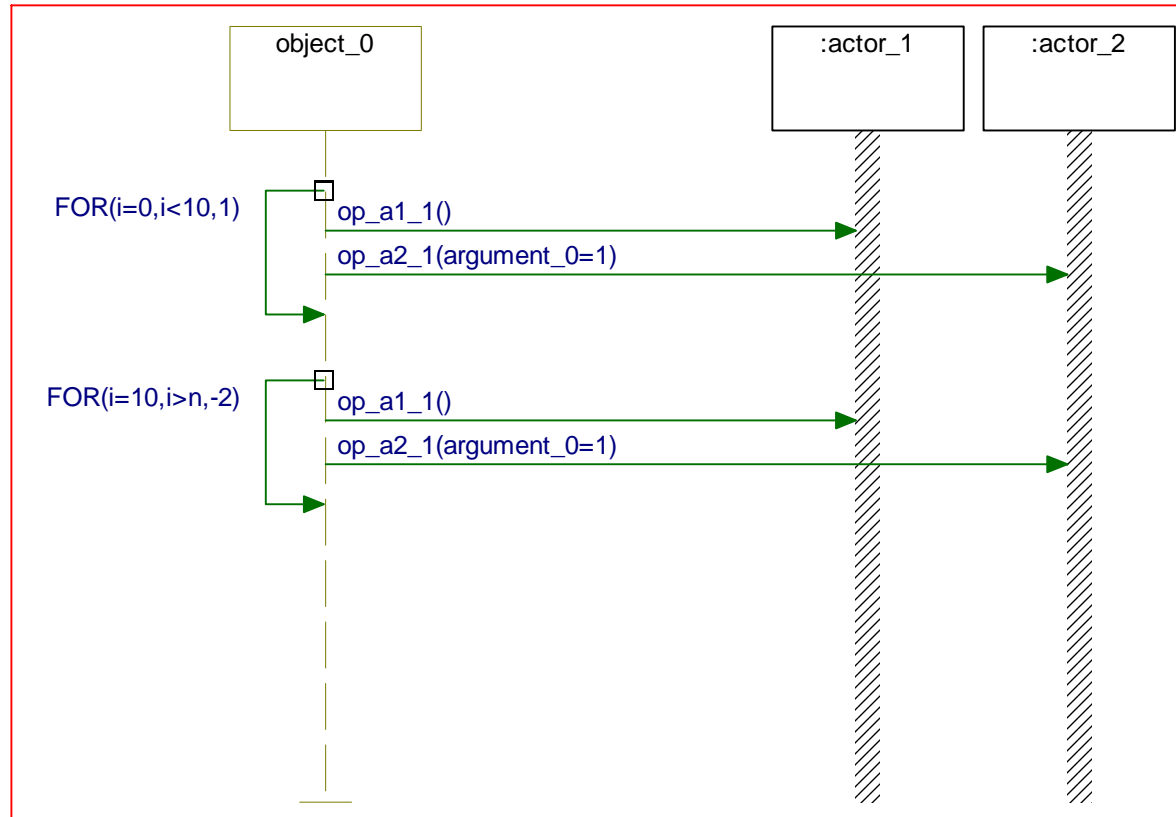
Operationen sind streng typisiert.

Operationen können mit oder ohne Argumente gerufen werden.

Operationen haben Rückgabewert oder nicht.

Rückgabewerte können nur Variablen sein.

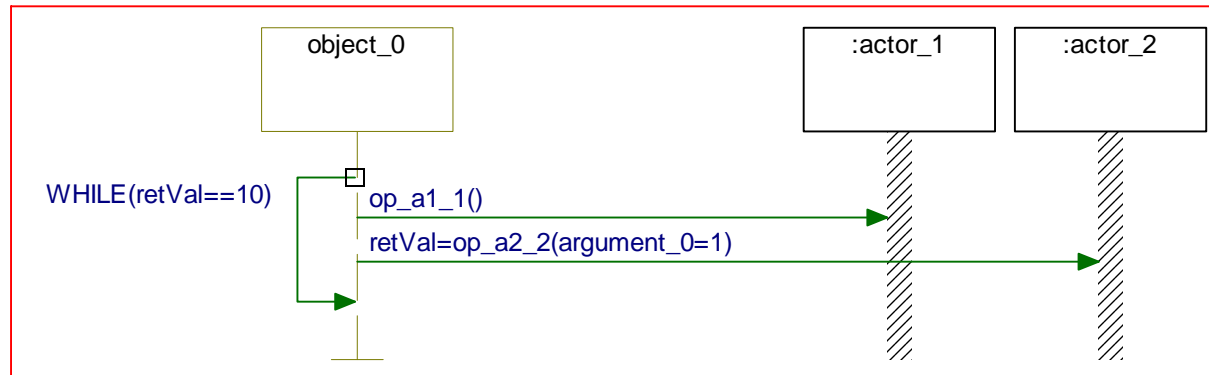
Argumente können Konstanten oder Variablen sein.



For-Schleifen umfassen einen Block.

Syntax wie in C: `FOR (i=0, i<10, 1)`
`FOR (i=10, i>0, -2)`

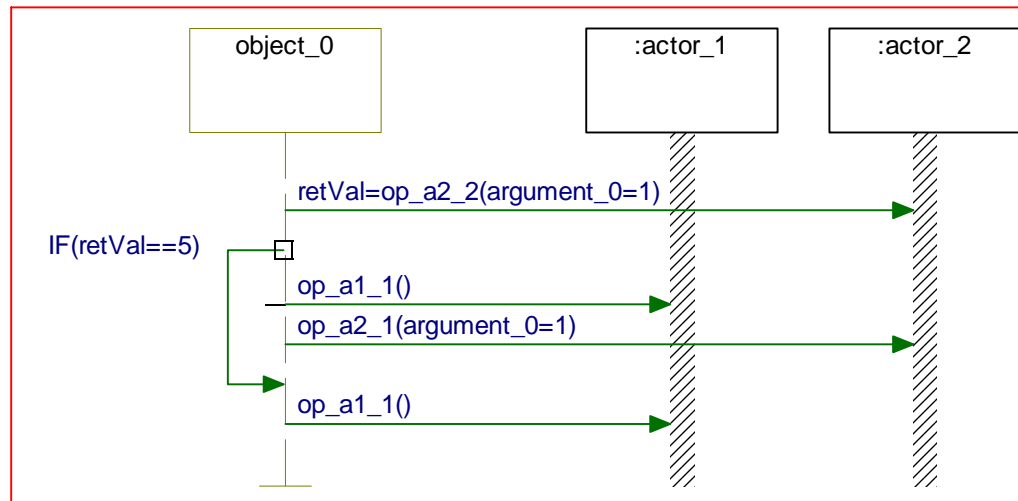
`for (int i=0; i<10, i++)`
`for (int i=10; i>0; i=i-2)`



While-Schleifen umfassen einen Block.

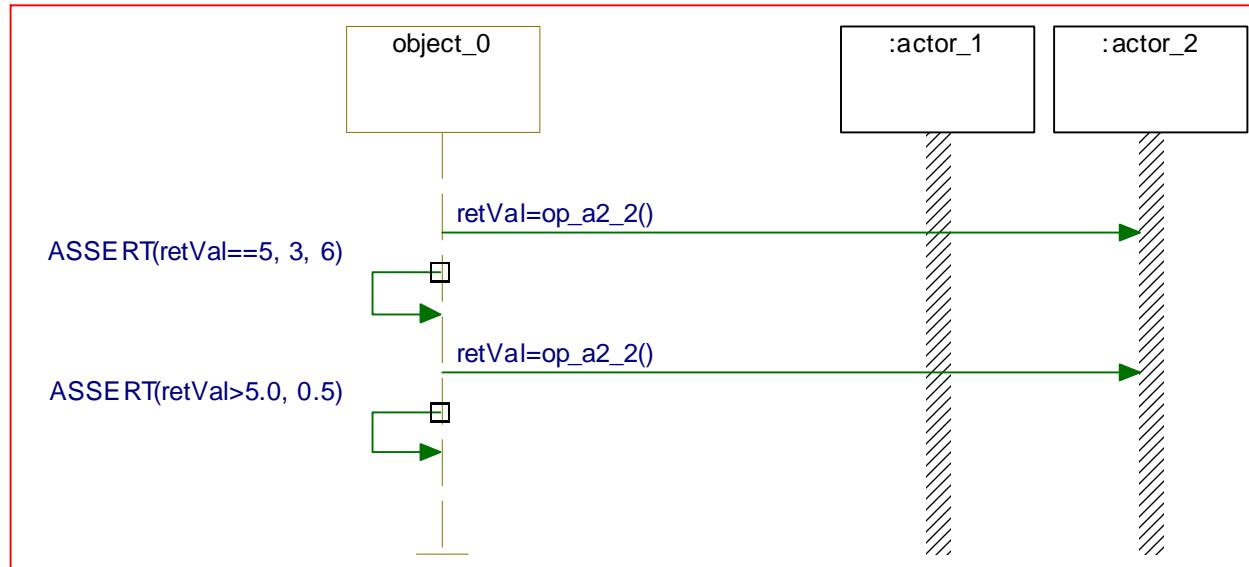
Syntax wie in C: `WHILE (retVal==10)`

`while (retVal==10)`



Eine if-Anweisung umfasst einen Block.

Es sind nur if-Zweige erlaubt, keine else Zweige.



Ein Assert ist ein Prüfung des Wertes mit Toleranzen.

Syntax: ASSERT (Bedingung, Min, Max)
 ASSERT (Bedingung, +/--Range)

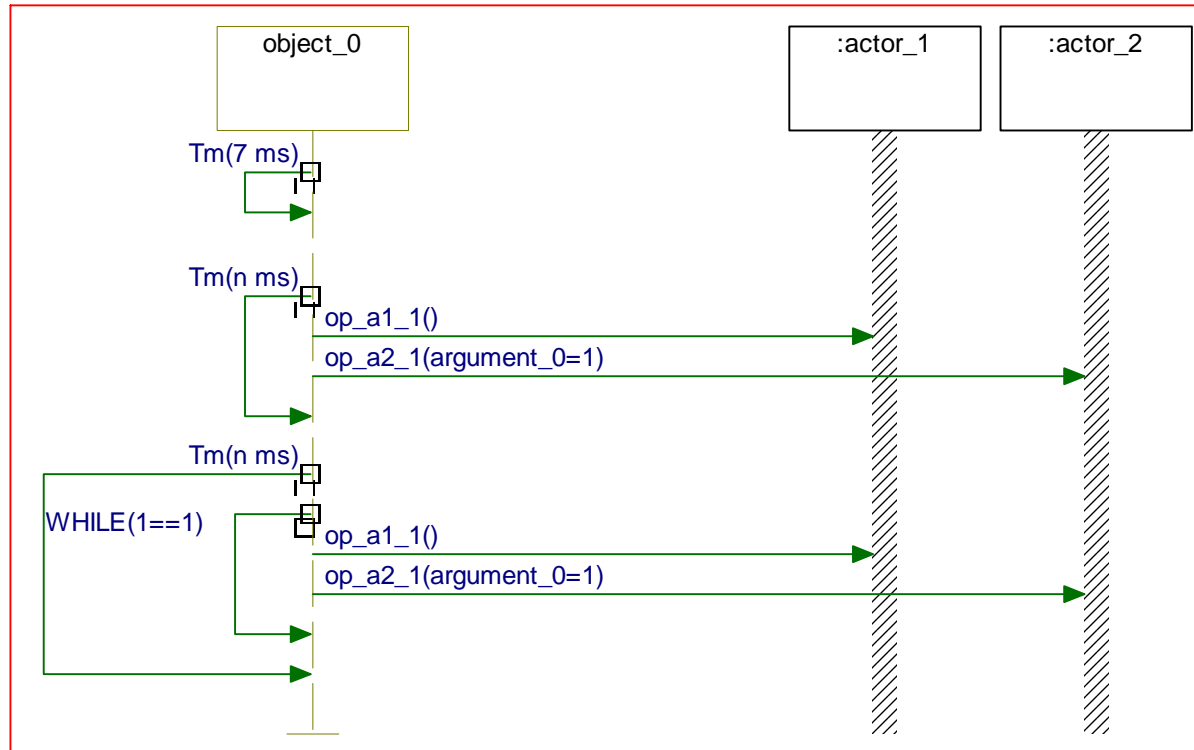
Bei misslungener Prüfung sofortiger Abbruch der Sequenz.

Der Testfall wird als Failed gekennzeichnet.

Verwendet werden dürfen alle Condition Ausdrücke.

Toleranzen beziehen sich immer auf die Grenzen:

- (retVal>5.0,0.5) meint retVal>4.5, (retVal<=6.0,0.5), retVal<=6.5
- (retVal!=5.0,0.5) meint retVal<4.5 and retVal>5.5



TM (Waits) ermöglichen ein genaues Warten oder Triggern.
Die Sequenz wird sofort unterbrochen und an der Wait-Marke (Pfeilspitze) fortgeführt.

Die Schritte nach Tm werden normal ausgeführt.

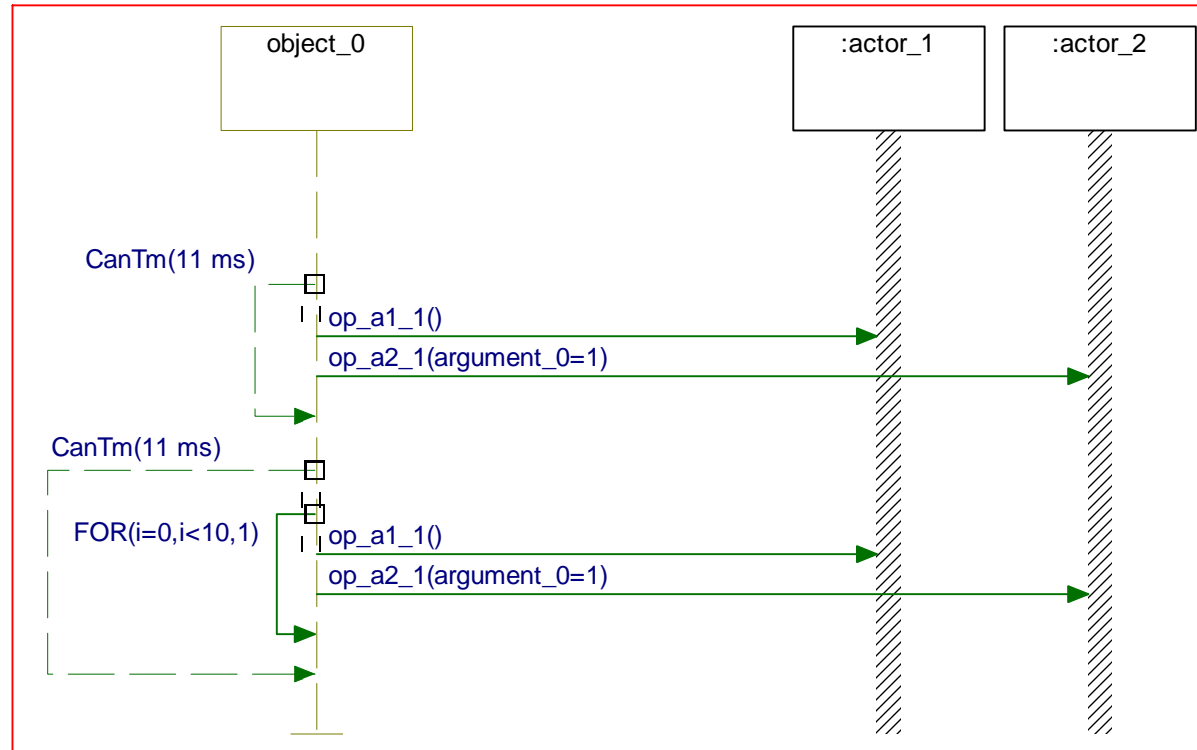
An der Pfeilspitze von Tm wird gewartet!

Eine Rückkehr ist nicht möglich.

Somit können auch Endlos-Schleifen unterbrochen werden.

Der 1. Timeout hat immer die höhere Priorität, deswegen (Start bei 1)

Zeiten können sein (s, ms, us, ns).



CanTm (Cancel Timer) ermöglicht einen genauen Watchdog.

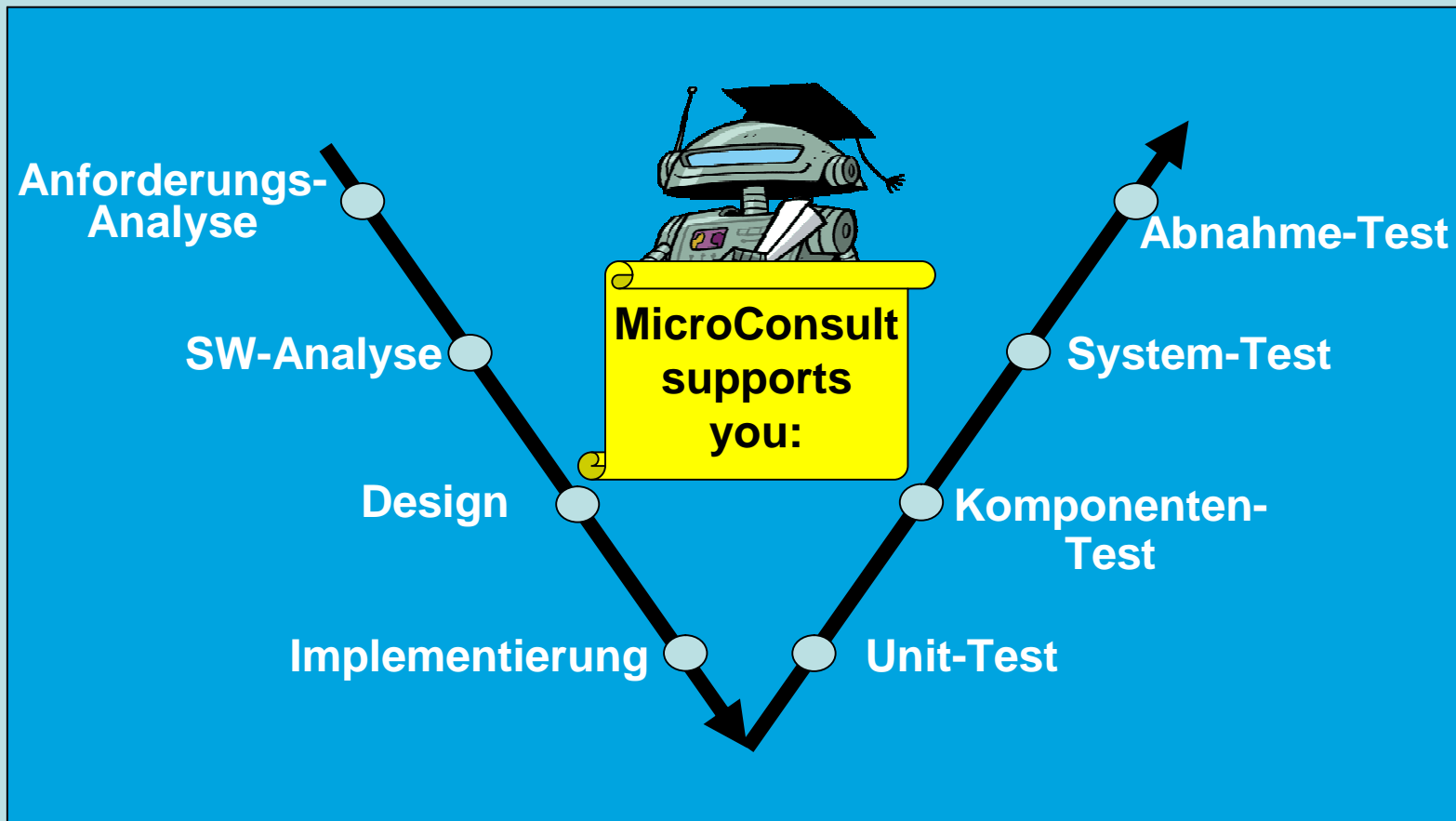
Die Sequenz wird normal weitergeführt.

Wird der Watchdog nicht innerhalb der Zeit zurückgesetzt, wird die Sequenz abgebrochen!

An der Pfeilspitze wird der Watchdog zurückgesetzt.

Der Testfall ist failed!

Consulting, Training, Workshops, Coaching, Projekte



HW-/SW-Technologien, Tools, Methoden, Prozesse, Team