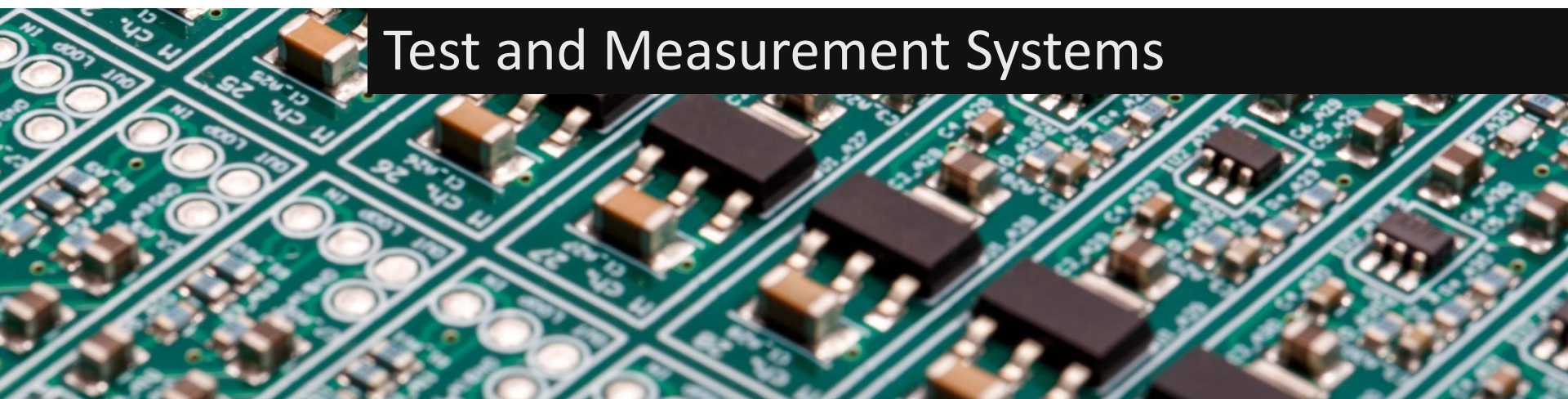




# ADDQ

**Question Everything!**<sup>TM</sup>



Test and Measurement Systems

# Presentation

- **Mattias Ericsson**

- LabVIEW developer ~15years
- CLA
- LabVIEW Partner Program

- QRM

- [www.addq.se/qrm](http://www.addq.se/qrm)

- G# Framework

- Free, open source tool
  - LabVIEW Add-On of the Year for Community 2011
  - [www.ni.com/labviewtools](http://www.ni.com/labviewtools)
  - [www.addq.se/gsharp](http://www.addq.se/gsharp)



# The Art of Archiceture

- What is a Artictecture?
- Conciderations
- Different Strategies
- Conclusions

# Architecture

- ” The **software architecture** of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships between them.”



*“Tell us what is important. Architecture is about the important stuff. Whatever that is.” – Ralph Johnson*



# Architecture – some aspects

- **Ralph Johnson** - *“The expert developers working on that project have a shared understanding of the system design. That shared understanding is called architecture.”*
- **Jim Coplien** - *“Architecture is when you design your abstract base classes.”*
- **Martin Fowler** – *“Two common elements: Highest-level breakdown of a system into parts and the decisions that are hard to change”*
- **Eoin Woods** – *“The set of design decisions which, if made incorrectly, will cause your project to be cancelled.”*



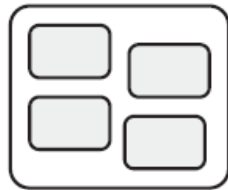
# Design vs Architecture

- What is the *difference*?
  - Architecture **is** design, but not all design is architecture.
  - Many design decisions are left unbound by the architecture.
  - Architecture is often referred as **strategic** design

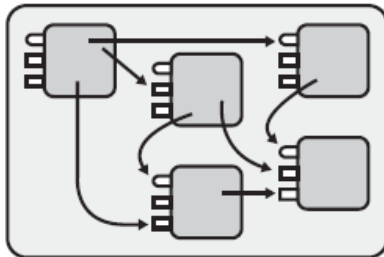
# Design Levels



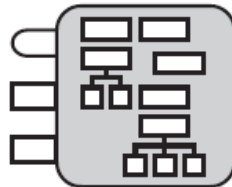
① Software system



② Division into subsystems/packages



③ Division into classes within packages



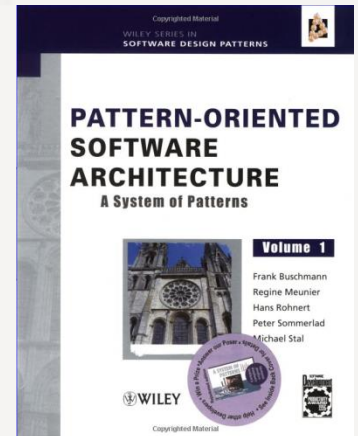
④ Division into data and routines within classes



⑤ Internal routine design

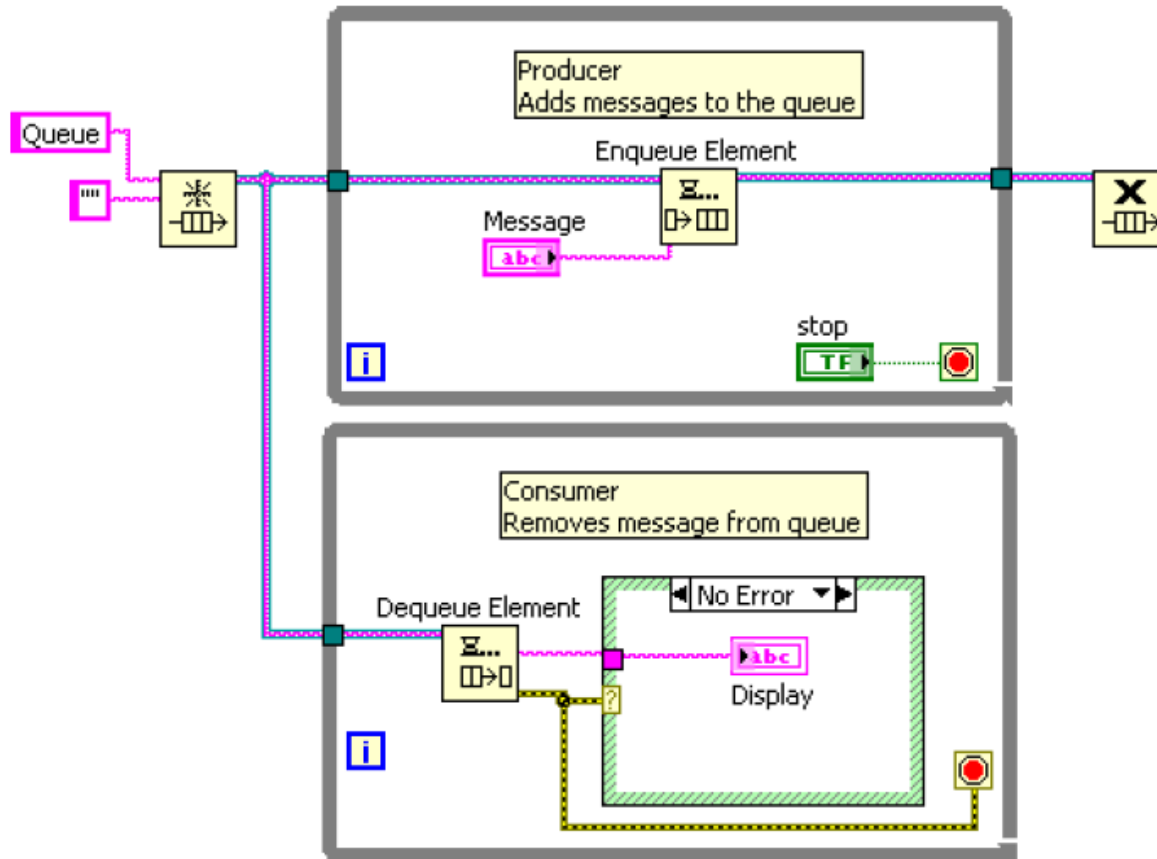
# Architectural Patterns

- Expresses a fundamental structural organization scheme for a software system
- Examples:
  - Model-View-Controller (MVC)
  - Layers
  - LabVIEW: Producer-Consumer (aka Master-Queue)





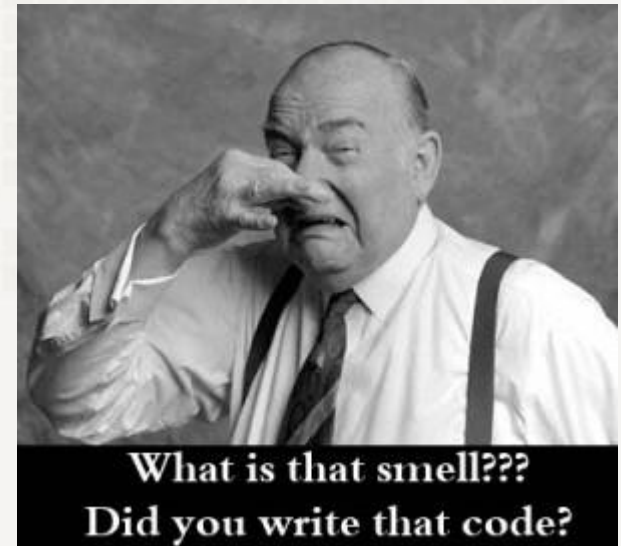
# Producer / Consumer



# Symptoms of Poor Design

## The Odors of Rotting Software

- Hard to change
- Easy to break
- Hard to reuse
- Hard to do the right thing
- Complexity and overdesign
- Repeating structures that could be unified under a single abstraction
- Hard to read and understand



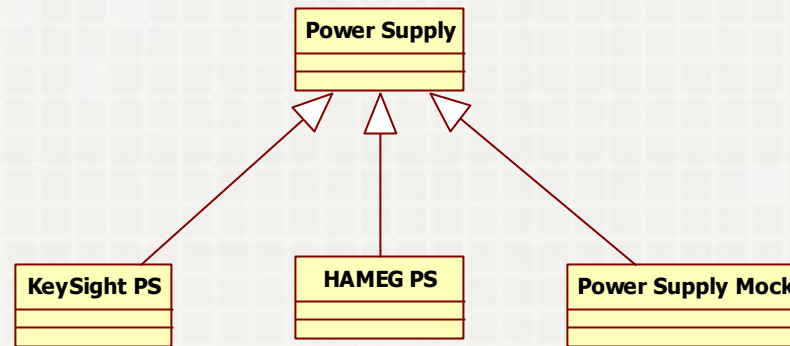
# Design Approches

- Top-Down
  - Your brain isn't force to deal with too many details at once
  - Decomposition strategy

- Bottom-up
  - Sometimes the top-down is so abstract it is hard to get started.
  - When identifying several low-level responsibilities, start from the top again!
  - Composition strategy



# Hardware Abstract Layer (HAL)



*To HAL or not to HAL is the question!*

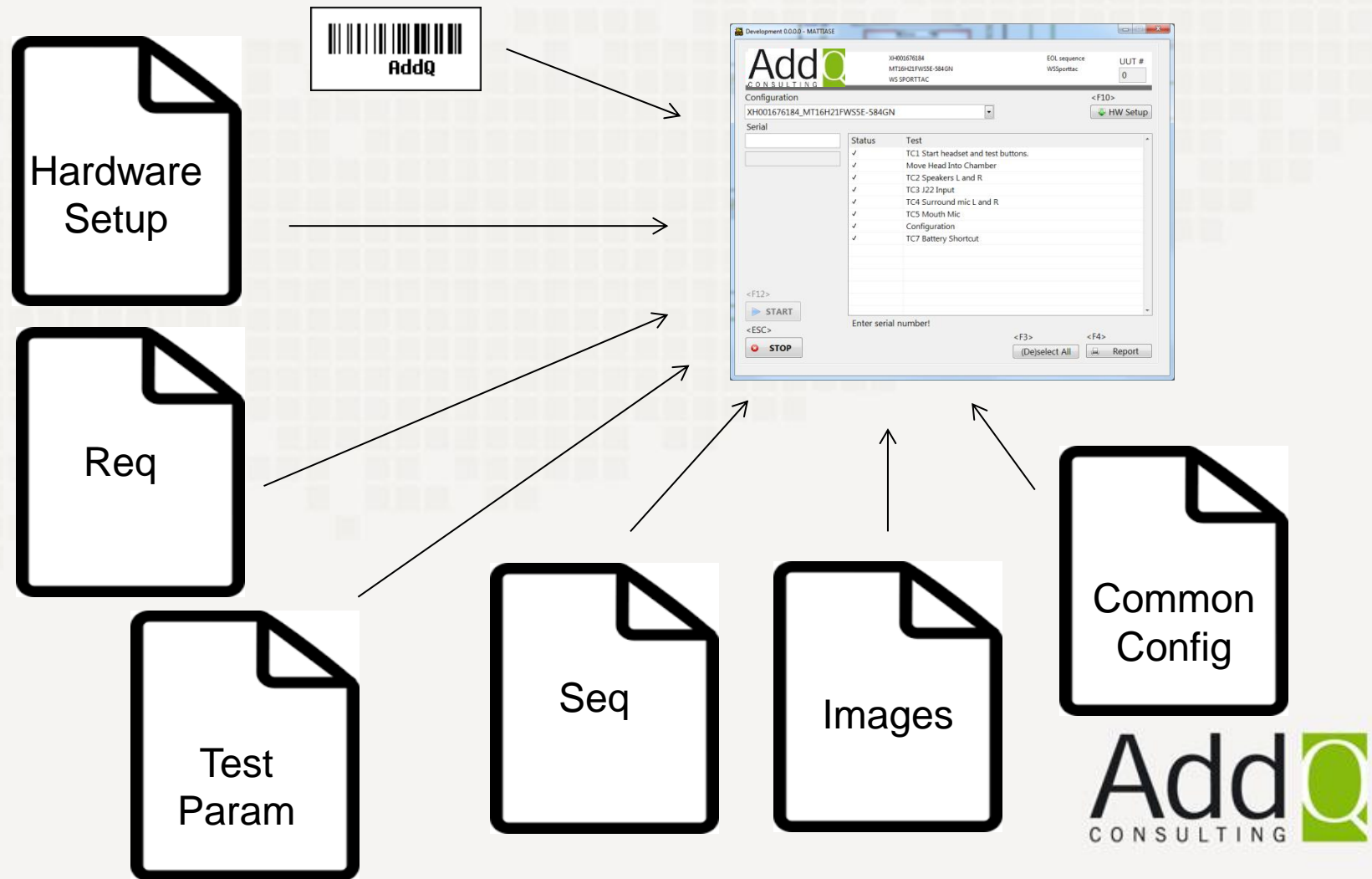


# Important Considerations

- Receiver of the system
  - Maintainer
  - Operator
- Station setups
  - Different brands of instruments
  - Different instrumentation
- Product support
  - Variants
  - Coexisting versions



# System Configuration



# Configuration Dependency

- Station Configuration
  - Hardware setup
  - COM ports etc.
- Product Configuration
  - Requirement and Sequences
  - Images
  - Different fixtures
    - Channel maps
  - Firmware
- Common Configuration
  - Platform setup

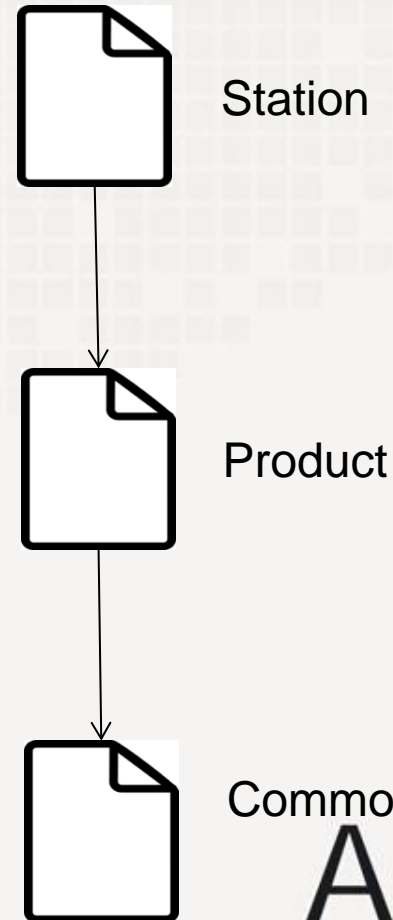


# Configuration Override

- INI-file
  - Search hierarchy
    1. File name
    2. Section
    3. Key
- XML
  - Compare Microsoft .NET



*Always use relative paths!*



# Example of Simplification

## Problem:

- TestStand
  - Sequences
  - Variants requires own sequence
- LabVIEW
  - Test Code State Machine – sometimes product specific
- INI for requirements
- INI for test cases
- Station specific
  - Channel maps
- No override for local changes

*Very hard for maintainer to know what to do!*

# Example of Simplification

Solution:

- LabVIEW
  - Test Code – generic simple steps
  - One platform for all station, products and variants.
- XML
  - Sequence
  - Requirements and parameters
  - Test steps
- Configuration override
  1. Station
  2. Product
  3. Common

*Easy for maintainer to manage!*



# XML

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<testsuite name="PCBA test" schemaVersion="1.0" xmlns="http://www.addq.se" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.addq.se test.xsd" autogenerateserial="false">
```

```
<uut autogenerateserial="false">
```

```
<setup>
```

```
<power>
```

```
<output>
```

```
<powerchannel channelname="MainSupply" voltage="14.45" currentlimit="3" enable="true">
```

```
</output>
```

```
</power>
```

```
<signalroute>
```

```
<checkroute logic="AND">
```

```
<condition name="EmergencyStop"/>
```

```
<evaluatetrue info="Please reset emergency stop!" type="QA"/>
```

```
</checkroute>
```

```
<route name="PowerEnable" set="true" waittime="100"/>
```

```
<route name="MainPowerSoftStart1" set="true" waittime="1200"/>
```

```
</signalroute>
```

```
</setup>
```

```
<test name="TC4.1" stopiffail="true">
```

```
<tdesc>Input current</tdesc>
```

```
<dmm>
```

```
<input>
```

```
<aichannel channelname="CurrentShunt">
```

```
<analysis method="amplitude" unit="A" comparator="GELE" reqmin="2.20" reqmax="2.20">
```

```
</aichannel>
```

```
<aichannel channelname="Sense">
```

```
<analysis method="amplitude" unit="V" comparator="GELE" reqmin="13.9" reqmax="13.9">
```

```
</aichannel>
```

```
</input>
```

```
</dmm>
```

```
</test>
```

```
<test name="TC4.2">
```

```
<tdesc>Over voltage turn off</tdesc>
```

```
<power>
```

```
<output>
```

```
<powerchannel channelname="MainSupply" voltage="18.0" currentlimit="3" enable="true">
```

```
</output>
```

```
</power>
```

```
<dmm>
```

```
<input>
```

```
<aichannel channelname="CurrentShunt">
```

```
<analysis method="amplitude" unit="mA" comparator="GELE" reqmin="0" reqmax="0">
```

```
</aichannel>
```

```
<aichannel channelname="Sense">
```

```
<analysis method="amplitude" unit="V" comparator="GELE" reqmin="17.9" reqmax="17.9">
```

```
</aichannel>
```

```
</input>
```

```
</dmm>
```

```
<power>
```

```
<output>
```

```
<powerchannel channelname="MainSupply" voltage="14.45" currentlimit="3" enable="true">
```

```
</output>
```

```
</power>
```

[Common]

Lock=DAQ1:DO, 300

ReversePolarity=DAQ1:DO, 301

MainPowerSoftStart1=DAQ1:DO, 305

MainPowerSoftStart2=DAQ1:DO, 306

DualPowerSupply=DAQ1:DO, 304

EmergencyStop=DAQ1:DI, 303::0

#Inverted

FixtureClosed=DAQ1:DI, 302::0

#Inverted

FixtureID=DMM:Resistance, Auto, N/A, 320

[UUT/1]

MainSupply=PS1

PowerEnable=DAQ1:DO, 400

# Port and channel

CurrentShunt=DMM:Voltage, 0, DC, 101

#Mode, range, bank, channel

OvertempResistor=DAQ1:DO, 000

LED1=LEDAn:1

LED2=LEDAn:2

LED3=LEDAn:3

LED4=LEDAn:4

LED5=LEDAn:5

LED6=LEDAn:6

Sense=DMM:Voltage, Auto, DC, 102

I-sense1=DMM:Voltage, Auto, DC, 103

I-sense2=DMM:Voltage, Auto, DC, 104

&&Common

[UUT/2]

MainSupply=PS1

PowerEnable=DAQ1:DO, 401

# Port and channel

CurrentShunt=DMM:Voltage, 0, DC, 105

#Mode, range, bank, channel

OvertempResistor=DAQ1:DO, 001

LED1=LEDAn:7

LED2=LEDAn:8

LED3=LEDAn:9

LED4=LEDAn:10

LED5=LEDAn:11

LED6=LEDAn:12

Sense=DMM:Voltage, Auto, DC, 106

I-sense1=DMM:Voltage, Auto, DC, 107

I-sense2=DMM:Voltage, Auto, DC, 108

&&Common

# Architectural Strategies

**KEEP  
IT  
SIMPLE  
STUPID**

# Architectural Strategies

- **What** to test?
  - Test object
  - Test equipment
- **Who** is going to maintain?
- Product and Variants
  - What's next to be tested?
- Serial Numbers
  - Generation
  - Scanning
- Documentation
  - Block diagrams e.g. UML
  - Schemes and Manuals
- Results
  - Measurements
  - Traceability





# Web Application

**QRM - Quality Result Management**

Overview Test Results Production Report Yield SPC

### Test Results

UUT Type Station Location Start Date Time Stop Date Time  
Q823 -All- -All- 2015-02-05 00:00:00 2015-02-11 23:59:00

UUT Serial Number UUT Batch UUT Setting Sequence  
-All- -All-

UUT Variant Result Set  
Execute -All- -All-

☐ Only Repaired

Presentation View CSV Headers  
By Result Set/Day Result Group + Result Name Download CSV

☐ Configuration Info ☐ Parameters ☐ Repairs ☐ Yield ☐ Sort Result Name

2015-02-09 (1)

- Q823\_PCBA\_Programming\_SingleComm\_B-PC\_#74 Start: 13:30:29 Stop: 14:00:16 (00:02:59) S
- S/N: 1335000113 Q823\_AVC07 Start: 13:57:17 Stop: 14:00:16 (00:02:59) S
- UUT Settings
  - TC328.1 NXP Gang Programming (TM02-001A) Loop: 0 Start: 13:57:19 Stop: 13:57:34 (00:00:15)
  - Q823.2.4.1 Bootloader Programming: True [True]
  - TC328.2 Flash Gang Programming (TM02-001A) Loop: 0 Start: 13:57:35 Stop: 13:58:12 (00:00:37)
  - Q823.2.3.2 Flash Programming: True [True]
  - TC328.3 Xmodem Programming (TM02-002A) Loop: 0 Start: 13:58:14 Stop: 13:59:10 (00:00:56)
  - Q823.2.4.3 Xmodem Programming: True [True]
  - TC328.4 Q823 Main Functionality (TM04-003A) Loop: 0 Start: 13:59:12 Stop: 13:59:59 (00:00:47)
  - Q823.2.1.2 Reverse Current: 0.20 mA [s=0.00; c=1.00]

**QRM - Quality Result Management** TestUser [Log Out]

Overview Test Results Production Report Yield SPC MSA Stations Repairs Export About

### Statistical Process Control (SPC)

UUT Type Station Location Start Date Time Stop Date Time Outcome Test Socket  
Q823 -All- -All- 2015-02-05 00:00:00 2015-02-11 23:59:00 -All- -All-

UUT Serial Number UUT Batch UUT Setting Sequence Test Type  
-All- -All-

UUT Variant Result Set Operator Test Program  
Execute -All- -All- -All- -All-

Result Name Result Group  
Q823.2.12.2 Play Voice Amp L -All-

Test Method  
☒ Only Numeric -All-

Max # of Samples 5000 Truncate Upper Window Start 0 Mode Index Remove Outliers  
Sub Group Size 1 Truncate Lower Window Stop 342 ☒ No Duplicate UUTs

Q823.2.12.2 Play Voice Amp L (V) (2015-02-05 12:14:15 -> 2015-02-09 13:59:33) Download CSV Download PDF ☐ Skip Histogram Limits

Individual Chart (UCL:0.160 LCL:0.150 Xbar mean:0.155)

Capability Histogram

Legend: within (solid red line), overall (dashed black line), LSL (0.1400) (dotted red line), USL (0.1800) (dotted red line)

# Background Meas Process

Test Executive - 0.0.0.0

Test Manual <F2>

<F5> <ESC> UUT # 1 CN Identity L90LS-06-13054-1 Serial Number 1

Section Test Time 00:00:07 Total Test Time 00:00:28 Block # 1 Section # 5

Start Stop

Pre Tests

Status	Test Cases
✓ PASSED	020 Anslutning
✓ RUNNING	050 Yttre läckage
✓	060 HTB fast
✓	080 Uppvärmning
✓	090 Inre läckage

Section Tests

Status	Test Cases
✓	300 OBE Address
✓	310 OBE Neutral Position
✓	370 OBE Parametrar Init
✓	100 Matarreducerare A
✓	100 Matarreducerare B
✓	340 OBE Hysteres A
✓	120 Tryckåterföring A
✓	380 OBE Kalibrera Start Flöde A
✓	130 Kapacitet A
✓	320 OBE Kalibrera Eset A
✓	325 OBE Verifiering Eset A
✓	330 OBE Kalibrera Power Float A
✓	335 OBE Verifiering Power Float A
✓	140 Tryckkompensering
✓	150 Lastberoende A
✓	160 Hysteres A
✓	350 OBE Tryckåterföring A
✓	360 OBE Flytläge A
✓	340 OBE Hysteres B
✓	120 Tryckåterföring B
✓	380 OBE Kalibrera Start Flöde B
✓	130 Kapacitet B
✓	320 OBE Kalibrera Eset B
✓	325 OBE Verifiering Eset B
✓	150 Lastberoende B
✓	160 Hysteres B
✓	350 OBE Tryckåterföring B
✓	100 MR slutinställning A
✓	100 MR slutinställning B
✓	165 Avlastning
✓	170 Slidläckage

Post Tests

Status	Test Cases
✓	065 HTB slutinställning

Section Status

5 ?

Filter A Filter B Filter T Filter FME Filter Pump Stop

Set pump pressure and flow. Deselect All

Set pressure to 320,0 bar. Report

TB Mode 3 FMA 0,8

TGP 51,1 FMB 0,1

TGA 21,1 FMM -29,2

TGB 95,9 FMT -31,2

TGT 12,9 FME -9,2

LSDIFF 23,1 Current 1

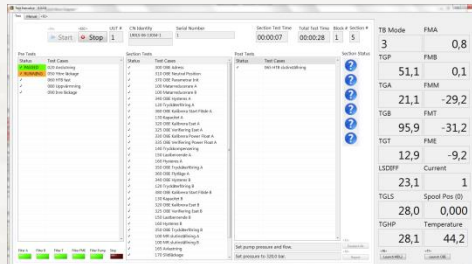
TGLS 28,0 Spool Pos (0) 0,000

TGHP 28,1 Temperature 44,2

<F6> Launch MDL2 <F7> Launch OBE



# Background Meas Process

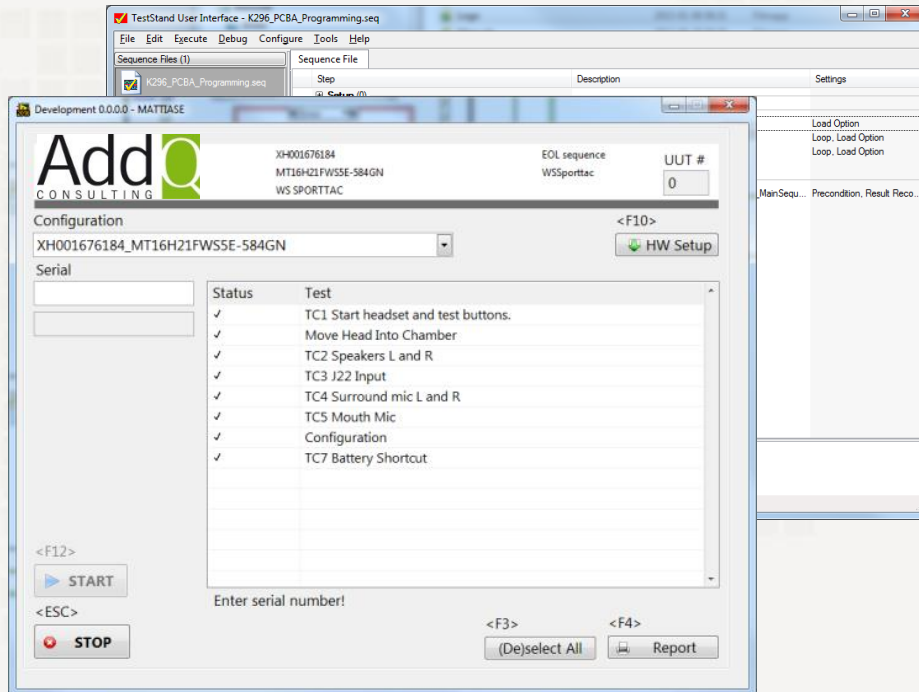


1. Application launch background process.
2. Background process measures and constantly writes to memory.
3. Application regular polls memory for latest meas values.
4. Application closes process when closing down.

- + Easy switch between Test and Manual
- + Always monitoring
- + Averaging
- Slow sampling



# "Hit and Run"



1. Application initialize equipment.
2. Sequential testing
  1. Measure
  2. Analyze
3. Application close equipment when closing down.

- + Easy control flow
- + Simple
- + High performance
- No monitoring

# Conclusion

- Architecture is not only software design
  - Configuration
  - Maintenance
  - Product and variants
  - Result Management
- No overdesign!
- Documentation
  - Keep up to date!
- KISS – Keep It Simple Stupid!