Test Equipment Specification for DPF1C with DS1 - Boxbuild and DPF1K with DS1 - Boxbuild

DPF1C

DPF1K

1. History

Date	Rev.	Changed by	Change(s) in details
2023-08-03	A	KB	New document started.
2023-08-14	B	KB	Corrected some error (content table, HW type)
2023-08-22	C	KB	Corrected some test step. Added a DPF1K to test also

2. Scope and purpose

This specification applies to the DPF1C with DS1 boxbuild test equipment.
This document describes the system interfaces, system events, and data in the system and system functions.
This document divides the system into subsystems with focus on data in the system, events in the system and function blocks.

3. Contents

4. System Design

4.1 General description

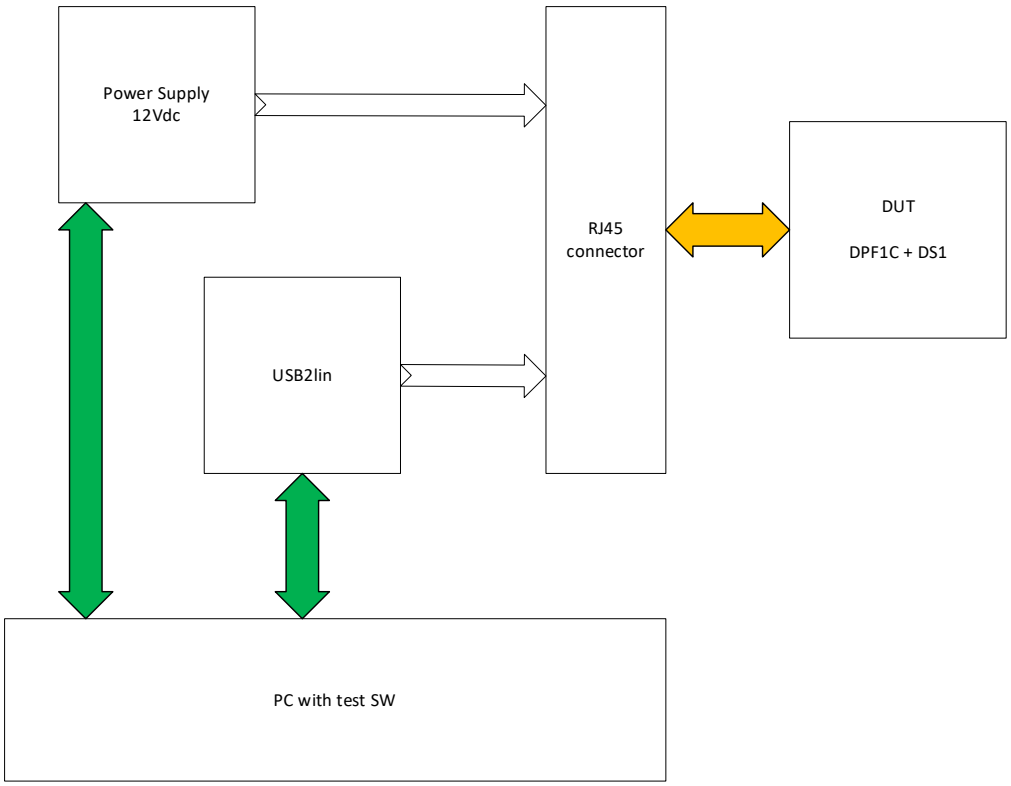
The device which has to be tested is a DPF1C with a DS1 or DFP1K with a DS1, and may in the following be referred to as a UUT (unit under test). The device will be used for LINAK Linear Actuators at tables.

The Boxbuild test equipment must test electrically that everything is mounted on the UUT, and the UUT must be fixed so that the UUT always has the same position during testing and can't be Shaked under test, otherwise the test result will fail in test step 1206 to 1208 and 2106 to 2108. It should be clearly indicated that the failing UUT easily can be identified as failing by red light.

4.1.1 Hardware

It is expected that the following hardware on the below block diagram will be used during test of the UUT, but other instruments and devices can be used.

4.1.1.1 Block Diagram



4.1.1.2 Test system

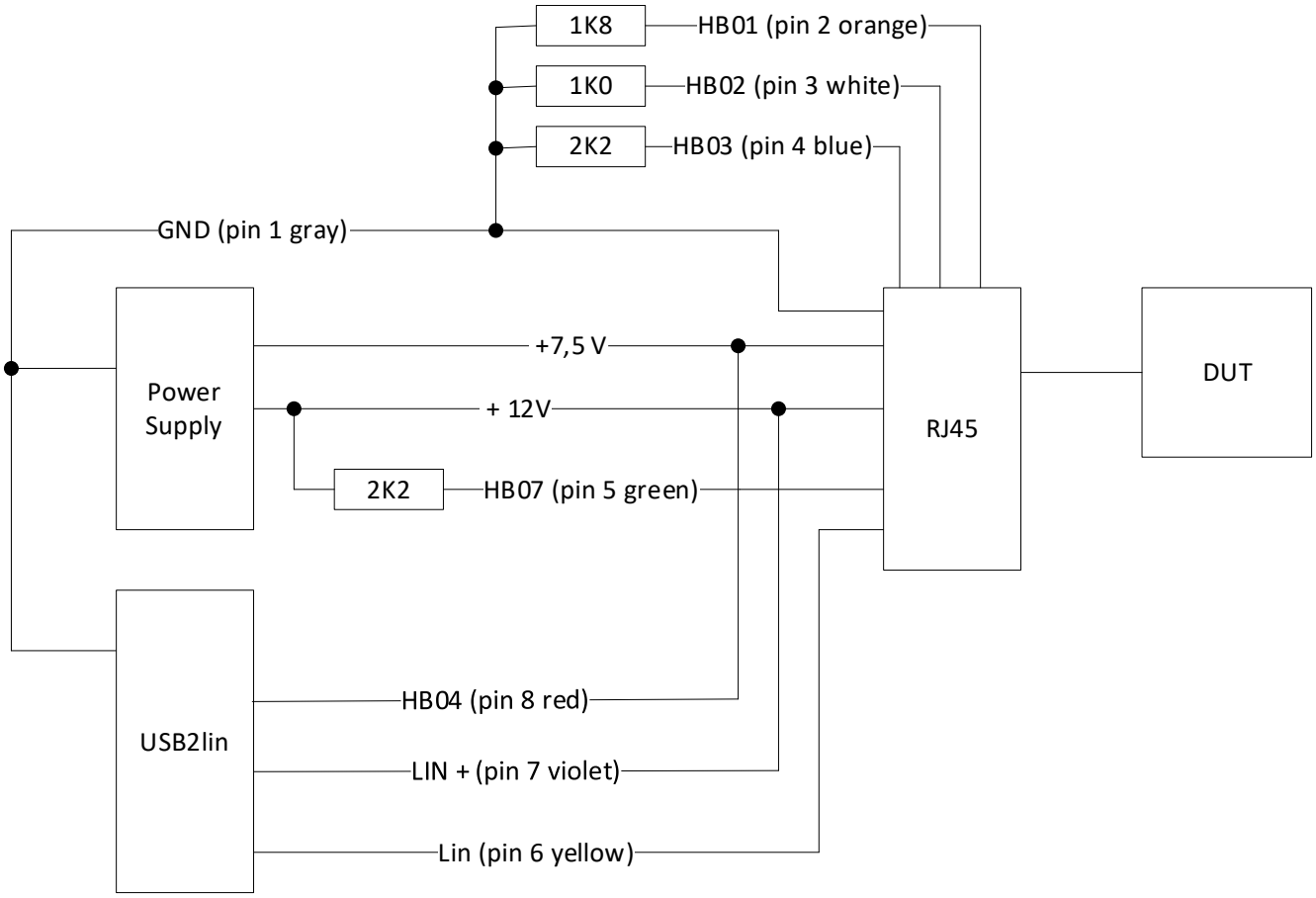
Test system with computer and measurement system.

The DC Power supply is used to provide supply voltage to the UUT. It is important that the Power supply can measure in 1 µA range, especially on the HB04 pin.

The USB2Lin is a LINAK device and will be provided by LINAK.

4.1.1.3 Connection of loads and power supply

The following schematic shows the necessary minimum setup for testing the boards. The resistor 1K8 on HB01 and 1K0 on HB02 are for future DPF1* variants.



4.1.1.4 Operating System

Windows 10 or newer must be used.

4.1.1.5 Test SW

The test sequence must be written in LabView.

4.1.1.6 LIMIT file

The LIMIT-file is written in a basic text-file. The file contains all MAX and MIN levels for all test steps.

4.1.1.7 Log

Measured data are stored in a LOG-file. See chapter 5.5 for more information.

4.2 Variants

It should also be simple to add new variants. In addition, it must be possible to scan a barcode instead of selecting a variant from a drop down menu. For each variant it must be possible to link some constant values.

These values are show below for the variant yyyDPF1Cyyyyy-xx-xx

[VARIANT]
yyyDPF1Cyyyyy-x-x
yyyDPF1Cyyyyy-x-x
yyyDPF1Kyyyyy-x-x

[BARCODE]
yyyDPF1Cyyyyy-x-xxxxxxxxxxxxxxxxxx
yyyDPF1Cyyyyy-x-xxxxxxxxxxxxxxxxxx
yyyDPF1Kyyyyy-x-xxxxxxxxxxxxxxxxxx

*x = A running number

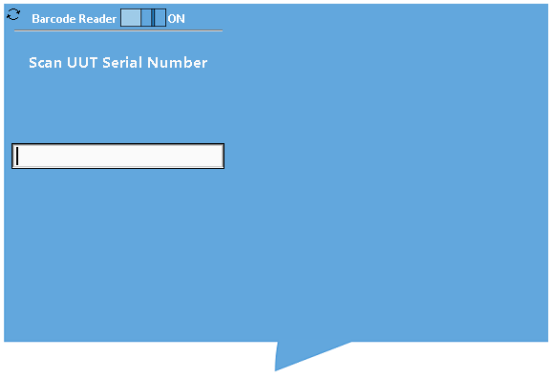
4.3 User interface

4.3.1 Barcode scanner disabled.

If the barcode scanner is disabled then the operator must choose the variant from a drop down menu manually. After that a start button must appear.

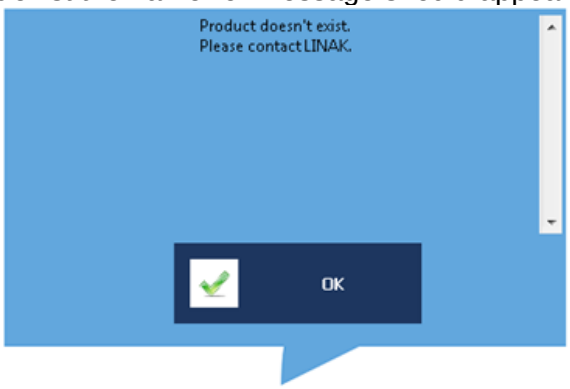
4.3.2 Barcode scanner enabled

If the barcode scanner is enabled then the drop down menu must be disabled and the variant auto selected from the barcode. It should now be possible to scan a Barcode instead.



Barcode is scanned every time a new UUT is tested. This means that the barcode number field need to have focus every time a test is completed.

In case the variant does not exist then an error message should appear:



5. Test execution

5.1 Starting Test

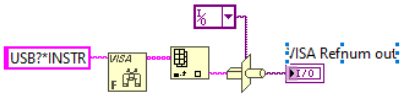
The test software must secure that Power supply is connected and that there is only one USB2lin connected. If those are not present or there is to many, then the test must stop, and the user must install the test equipment correctly, and hereafter restart the test software.

For USB2lin:

Send the USB2lin DLL command: u2l_FindAllLinakDevices (uint8 *DevArr, uint16 MaxLen)
On function return gives number of connected USB2lin.

You can get a VI from LINAK, for this command.

For Power Supply:



Result will give you an address for all power supply that are connected.

5.2 Start test Sequens

The test starts when the fixture is closed and a variant is selected or the barcode is scanned.
Send the USB2lin DLL command u2l_CloseDevice. You can get a VI from LINAK, for this command.

5.3 Test setup for DPF1C+DS1

Send the USB2lin DLL command u2l_CloseDevice
Return = 1 = success

5.4 Initial test

Send USB2lin DLL command u2l_OpenFirstDevice. (You can get a VI from LINAK, for this command)
Return = 1 = success
Send the USB2lin DLL command u2l_SetUsageOfControlLink(uint8_t ControlLinkUsage)

Input for DLL command: ControlLinkUsage = 4 (You can get a VI from LINAK, for this command)
Return = 1 = success

Set +12V on Lin + with limit on 20mA and 7,5V on HB04 with limit on 10mA, Wait 250ms.

Measure the DC current for LIN+ (test no. **1001**).

Measure the DC current for HB04 (test no. **1002**).

Turn off the voltage for Lin+, Wait 500ms.

Measure the DC sleep current for HB04 (test no. **1003**).

5.4.1 Communication test

Turn off the voltage for HB04. Wait 100ms. Send the USB2lin DLL command u2l_SetUsageOfControlLink(uint8_t ControlLinkUsage).
Input for DLL command: ControlLinkUsage = 3
Return = 1 = success

*Read S/N. send the USB2lin DLL command lin_RC_GetProductionInfo(uint8_t LinAddr, CStr SerNoString, CStr DateString), (*You can get a VI from LINAK, for this command).
Return = 0 = success

Input for DLL command: LinAddr = 16, SerNoString = empty, DateString = empty

Read S/N for DS1 (test no. **1201**).

Read Production date for DS1 (test no. **1202**).

Send the USB2lin DLL command lin_RC_StartTestMode(uint8_t LinAddress, uint8_t *Data, uint8_t DataSize). (You can get a VI from LINAK, for this command).
LinAddr = 136, Data = empty, Datasize = 0.

Send the USB2lin DLL command lin_RC_GetTestModeResult(uint8_t LinAddress, CStr Data, uint8_t MaxDataSize). (You can get a VI from LINAK, for this command).
LinAddr = 136, Data = empty, MaxDatasize = 200.
Result is a string, convert the string to a 8 bit array, hi byte is placed in the first and low byte in the 2.

For Gyro X result = Add Byte 1 and Byte 2 to a 16 bit answer (test no. **1203**).

For Gyro Y result = Add Byte 3 and Byte 4 to a 16 bit answer (test no. **1204**).

For Gyro Z result = Add Byte 5 and Byte 6 to a 16 bit answer (test no. **1205**).

For Acc X result = Add Byte 7 and Byte 8 to a 16 bit answer (test no. **1206**).

For Acc Y result = Add Byte 9 and Byte 10 to a 16 bit answer (test no. **1207**).

For Acc Z result = Add Byte 11 and Byte 12 to a 16 bit answer (test no. **1208**).

*Read S/N send the USB2lin DLL command lin_RC_GetProductionInfo(uint8_t LinAddr, CStr SerNoString, CStr DateString), (*You can get a VI from LINAK, for this command).

Input for DLL command: LinAddr = 8, SerNoString = empty, DateString = empty

Read S/N for DPF1C (test no. **1209**).

Read Production date for DPF1C (test no. **1210**).

5.4.2 Programming CFG file

Send the USB2lin DLL command lin_OpenConfig(CStr HardwareType, uint8_t HexFileName, int32_t MaxLen). (You can get a VI from LINAK, for CFG programming).
Input for DLL command: HardwareType = DS1, HexFileName = 0, MaxLen = 1032
Answer = Return = 0 = Error

Send the USB2lin DLL command lin_LoadConfig(CStr FileName, CStr HexFileName, int32_t MaxLen)
Input for DLL command: Filename = Path to CFG file, HexFileName = NULL, MaxLen = 0
Answer = Return = 0 = OK

Send the USB2lin DLL command lin_ProgramConfig(int32_t OnlyModified)
Input for DLL command: OnlyModified = 0
Answer = Return = 0 = OK (test no. **1301**).

Send the USB2lin DLL command u2l_CloseDevice
Return = 1 = success

5.5 Test setup for DPF1K+DS1

Send the USB2lin DLL command u2l_CloseDevice
Return = 1 = success

5.6 Initial test

Send USB2lin DLL command u2l_OpenFirstDevice.
Return = 1 = success
Send the USB2lin DLL command u2l_SetUsageOfControlLink(uint8_t ControlLinkUsage)
Input for DLL command: ControlLinkUsage = 4
Return = 1 = success

Set +12V on Lin + with limit on 20mA and 7,5V on HB04 with limit on 10mA, Wait 250ms.

Press button up

Measure the DC current for button up (LIN+) (test no. **2001**).

Measure the DC current for button down (HB04) (test no. **2002**).

Release press on button up. Hereafter Press button down. Wait 200ms.

Measure the DC current for button down (LIN+) (test no. **2003**).

Measure the DC current for button up (HB04) (test no. **2004**).

Release press on button down

Measure the DC current for LIN+ (test no. **2005**).

Measure the DC current for HB04 (test no. **2006**).

Turn off the voltage for Lin+, Wait 500ms.

Measure the DC sleep current for HB04 (test no. **2007**).

5.6.1 Communication test

Turn off the voltage for HB04. Wait 100ms. Send the USB2lin DLL command u2l_SetUsageOfControlLink(uint8_t ControlLinkUsage).
Input for DLL command: ControlLinkUsage = 3
Return = 1 = success

Read S/N. send the USB2lin DLL command lin_RC_GetProductionInfo(uint8_t LinAddr, CStr SerNoString, CStr DateString).
Return = 0 = success

Input for DLL command: LinAddr = 16, SerNoString = empty, DateString = empty

Read S/N for DS1 (test no. **2101**).

Read Production date for DS1 (test no. **2102**).

Send the USB2lin DLL command lin_RC_StartTestMode(uint8_t LinAddress, uint8_t *Data, uint8_t DataSize).
LinAddr = 136, Data = empty, Datasize = 0.

Send the USB2lin DLL command lin_RC_GetTestModeResult(uint8_t LinAddress, CStr Data, uint8_t MaxDataSize).
LinAddr = 136, Data = empty, MaxDatasize = 200.
Result is a string, convert the string to a 8 bit array, hi byte is placed in the first and low byte in the 2.

For Gyro X result = Add Byte 1 and Byte 2 to a 16 bit answer (test no. **2103**).

For Gyro Y result = Add Byte 3 and Byte 4 to a 16 bit answer (test no. **2104**).

For Gyro Z result = Add Byte 5 and Byte 6 to a 16 bit answer (test no. **2105**).

For Acc X result = Add Byte 7 and Byte 8 to a 16 bit answer (test no. **2106**).

For Acc Y result = Add Byte 9 and Byte 10 to a 16 bit answer (test no. **2107**).

For Acc Z result = Add Byte 11 and Byte 12 to a 16 bit answer (test no. **2108**).

5.6.2 Programming CFG file

Send the USB2lin DLL command lin_OpenConfig(CStr HardwareType, uint8_t HexFileName, int32_t MaxLen).
Input for DLL command: HardwareType = DS1, HexFileName = 0, MaxLen = 1032
Answer = Return = 0 = Error

Send the USB2lin DLL command lin_LoadConfig(CStr FileName, CStr HexFileName, int32_t MaxLen)
Input for DLL command: Filename = Path to CFG file, HexFileName = NULL, MaxLen = 0
Answer = Return = 0 = OK

Send the USB2lin DLL command lin_ProgramConfig(int32_t OnlyModified)
Input for DLL command: OnlyModified = 0
Answer = Return = 0 = OK (test no. **2201**).

Send the USB2lin DLL command u2l_CloseDevice
Return = 1 = success

5.7 Cycle time

The cycle time for the test of DUT without handling has to be at the max. of 10 seconds or below.

5.8 Data logging

Data from all tests IDs must be stored in a logfile. The file must be in .CSV format and a new file must be established every week. The file name could be named “Variant - Year – week number – computername”.

5.8.1 Format

The following information must be logged in the file: PO Number, Barcode, Timestamp (real-time for the measurement), Test step number, Upper limit, Lower limit, Measured value, Unit, Passed/Failed

5.8.2 Logging

A line in the LOG-file must indicate whenever the test is started and test is finished. If all test steps successfully finished, and the board are ok, a line that says “Test completed successfully” must be placed.

If a test fails, the text “Test failed” must be placed right after the Test number that failed.

5.8.3 Example of a LOG-file

An example of the log file database is available upon request.

6. Test ID list

The following tables show the limits for the different variants. Low- and High limit points out the max. and min. limit for the actual measurement. When Low and High limits are the same there is only one correct value.

6.1 Limits for yyyDPF1Cyyyyyyyy-x-x

No	Limit Name	Target	Low Limit	High Limit	Unit	Oracle Code
1001	DC current for Lin+	16	12,00	20,00	mA	CURRENT
1002	DC current for HB04	1	0,50	4,00	mA	CURRENT
1003	DC sleep current for HB04	17	10	60	µA	CURRENT
1201	Read S/N for DS1		Not empty		Str	STRING
1202	Read production date for DS1		Not empty		Str	STRING
1203	Gyro X	5600	5100	6100	INT	INTEGER
1204	Gyro Y	5150	4650	5650	INT	INTEGER
1205	Gyro Z	5850	5250	6450	INT	INTEGER
1206	Acc X	8500	7900	9100	INT	INTEGER
1207	Acc Y	7850	7250	8450	INT	INTEGER
1208	Acc Z	23000	21000	25000	INT	INTEGER
1209	Read S/N for DPF1C		Not empty		Str	STRING
1210	Read production date for DPF1C		Not empty		Str	STRING
1301	Programming CFG file for DS1	1	1	1	Bol	BOLEAN

6.2 Limits for yyyDPF1Kyyyyyyy-x-x

No	Limit Name	Target	Low Limit	High Limit	Unit	Oracle Code
2001	DC current for button up	12,72	12,22	13,52	mA	CURRENT
2002	DC current for button down	0	-1,00	1,00	mA	CURRENT
2003	DC current for button up	0	-1,00	1,00	mA	CURRENT
2004	DC current for button down	3,4	2,90	3,90	mA	CURRENT
2005	DC current for Lin+	10	5	15	mA	CURRENT
2006	DC current for HB04	1	0,5	4	mA	CURRENT
2007	DC sleep current for HB04	17	10	60	µA	CURRENT
2101	Read S/N for DS1		Not empty		Str	STRING
2102	Read production date for DS1		Not empty		Str	STRING
2103	Gyro X	5600	5100	6100	INT	INTEGER
2104	Gyro Y	5150	4650	5650	INT	INTEGER
2105	Gyro Z	5850	5250	6450	INT	INTEGER
2106	Acc X	8500	7900	9100	INT	INTEGER
2107	Acc Y	7850	7250	8450	INT	INTEGER
2108	Acc Z	23000	21000	25000	INT	INTEGER
2201	Programming CFG file for DS1	1	1	1	Bol	BOLEAN

Future target value for DPF1M
M1 = 12/(2K2+1K) = 3,75 mA, M2 = 7,5/1K = 7,50 mA
M3 = 12/(2K2+1K8) = 3,00 mA, S= 7,5/1K = 21,43 mA