Software
Requirements Specification

For

4 x Gen4L-block Tester software

Version 07

	Name	Date
Written by:	Meir Bar Nathan	
Approved by:		
Version No:	Version 02	5-11-2013
	Version 03	13-11-2013
	Version 04	18-11-2013
	Version 05	02-02-2014
	Version 06	09-02-2014
	Version 07	16-02-2014

- **Overview**

The Gen4L tester program will allow a comprehensive testing of the Gen4L-based print-block. The hardware configuration of the tester will be based on the electronics designed for the printer, namely, OHDB 2 + GEN4L Head driver Board. Some changes to the OHDB firmware may be needed in order to support the tester functionality.
The tester program can be based on the existing Eden Tester software which will undergo changes emanating from the requirements presented in this document.

- **Tester functions**

.1. **Temperature control**

.1.1. Set temperature for 4 ptintheads, each containing ODD and EVEN heaters (8 heaters altogether). The temperature set value will be entered in ADU.

.1.2. Set temperature for 4 channel block heaters (pre-heater1/2, block front/rear).

.1.3. Actual printheads and block temperature reading display in ADU (8+4 channels)

.1.4. Set error margin. The error margin will be defined in ADU (e.g. 100). The margin includes 2 parameters – Low limit and High limit. If the actual temperature value (in ADU) > High or < Low, the nackgroung of the relevant actual value will turn red. Otherwise it will be green.

.1.5. Heater ON/OFF control (global).This button enable/disable the heating of all channels.

.1.6. Individual Enable / Disable control for each heater.

.1.7. Temperature ramp-up control. The ramp-up of the temperature should ensure that the head temperature should track the block temperature. Once the block has reached the target temperature the controls become independent. See flow chart in Appendix A (warmup). Please note that the conversion of ADU to temperature uses 2 different formulae (one for the block thermistor reading and one for the head thermistor reading. The calculation process will include the following steps:

.1.7.1. The block temperature will be calculated by the average of all the block thermistors. The first formula will calculate the temperature of the block on all block thermistors and average the results.

.1.7.2. The resulting average will be entered into the second formula and the ADU that will be used for setting the heads' temperature will be calculated.

.1.7.3. Formula 1 (block temperature calculation):

$$T(block) = 119.94 \times 2.7183\text{^}(-8*10\text{^}-4 \times ADU)$$

Formula 2 (Heads ADU calculation):

$$ADU\ (heads) = 0.0025 \times T^3 - 0.3763 \times T^2 - 14.919 \times T + 3714.1$$

.1.8. Safeguards

.1.8.1. All head masks should be removed.

.1.8.2. If one of the thermistors was found faulty (short or disconnect), all heaters will be disabled.

.2. Data setting

.2.1. Individul nozzle setting for each of the 4 printheads (4 x 384 nozzles).
.2.2. Set ALL/Clear ALL control for all printheads.
.2.3. Set/Clear data for a group of nozzles (GUI dependent).
.2.4. Load/Save pattern capability.
.2.5. Cyclic data generation (cyclic ON/OFF pattern).

.2.5.1. Parameters for cyclic data:

1. Start nozzle
2. End Nozzle
3. # ON, #OFF

.3. Strobe LED control

The setting of the delay between the Fire pulse and the LED drive-pulse can be manually and interactively controlled by the user.
The GUI will be based on a slider as shown in the picture below:

200µS

0µS

Min delay = 0µs
Max delay= 200µs
Resolution = 1µs
The parameter sent to the firmware of the OHDB will be calculated by the following formula: *Delay time /T clock cycle*

A second parameter that should be controlled is the Strobe Duration. It will be defined by the user in units of µS. The parameter that will be sent to the firmware of the OHDB will be calculated by the following formula: *Strobe Duration /T clock cycle*

.4. Fire control

.4.1. Fire frequency setting. A sanity check should be done, based on the fire pulse design. In case of a violation (frequency higher than the maximal frequency) , a pop-up message wil appear noting the maximal frequency allowed.
The max frequency is calculated as follows:

$$Max\_frequency\ (single\ pulse) = 1/(Pulse\_width + interval) \times 10^6$$
$$Max\_frequency\ (double\ pulse) = 1/(Pulse\_width1 + Pulse\_width2 + 2 \times interval) \times 10^6$$

.4.2. Fire Mode setting
.4.2.1. Continuous
.4.2.2. Duty cycle (# of fires ON, # of fires OFF, # of cycles)
.4.2.3. Single burst (# of fires)
.4.2.4. Firing for a set time duration
Formula for conversion of time to # of fires:

# of fires = time (sec) x frequency (Hz)

.5. Head voltage control
.5.1. Calibration of head voltage (4 x 2 channels) (see appendix B)
.5.2. Actual voltage reading
.5.3. Error indication
.5.4. Heads PS voltage reading

.6. Resin-fill control

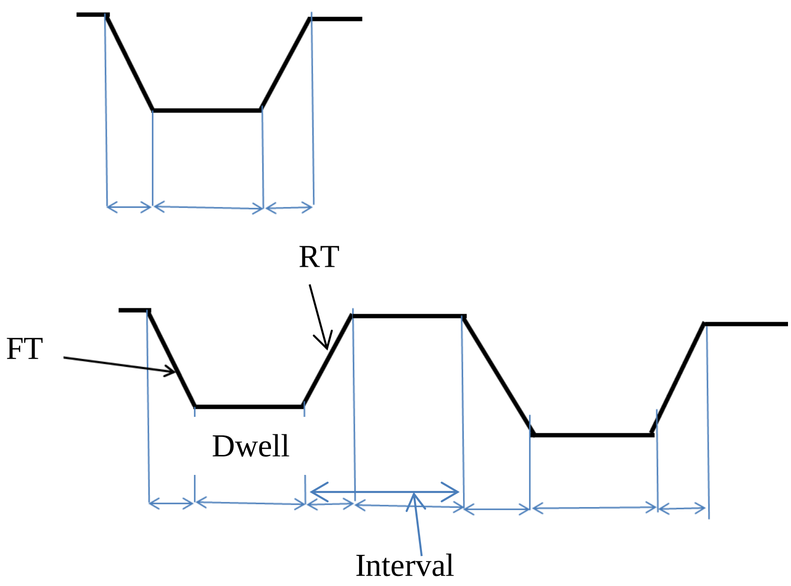
.6.1. Setting threshold values for all 4 sensors or for all 6 sensors in case where a "flooding" reservoir is being used.
.6.2. Setting hysteresis value
.6.3. Enable/disable control
.6.4. Setting timeout period
.6.5. Timeout error indicator
.6.6. Pump Active indicator for every one of the materials
.6.7. Setting of operating mode. Select between "flood" mode and 4-material mode. In case of a"flood" mode – selection of material source (pump assignment).

.7. Vacuum and purge control

.7.1. Vacuum valve On/Off control
.7.2. Setting purge duration
.7.3. Purge On/Off control
.7.4. Air Valve actuator
.7.5. Actual vacuum sensor readout

.8. Fire pulse builder

The Pulse Builder is a tool for setting the pulse parameters. It will be capable of handling single and double pulse configurations. 8 sets of parameters will be sent to the H/W (1 set for each half of the printhead).



- **Pulse parameters**

- **Single Pulse**
  - For a single pulse only the following parameter should be user defined:
    - ***Pulse width*** (user defined through GUI). The pulse width is defined as the time interval between the start of the falling edge and the start of the rising edge.

- **Double pulse**

In this mode the calculation of the dwell time will be done automatically depending on the the voltage and pulse width. Those parameters will be sent to the H/W after calculation.

Pulse width(µs) = User entry
Dwell time (µs) = *Pulse width - Tdelay*

- ***% of voltage*** (the voltage that has been calibrated) – for double-pulse only, and only for the first pulse (the second pulse will always be at full voltage).

Another parameter that will be required by the firmware is the lapse time between the start of the pulse and the trancation point. It will be calculated using the following formula:

Tdelay (µs) = (*Voltage x % of voltage / slew rate)*

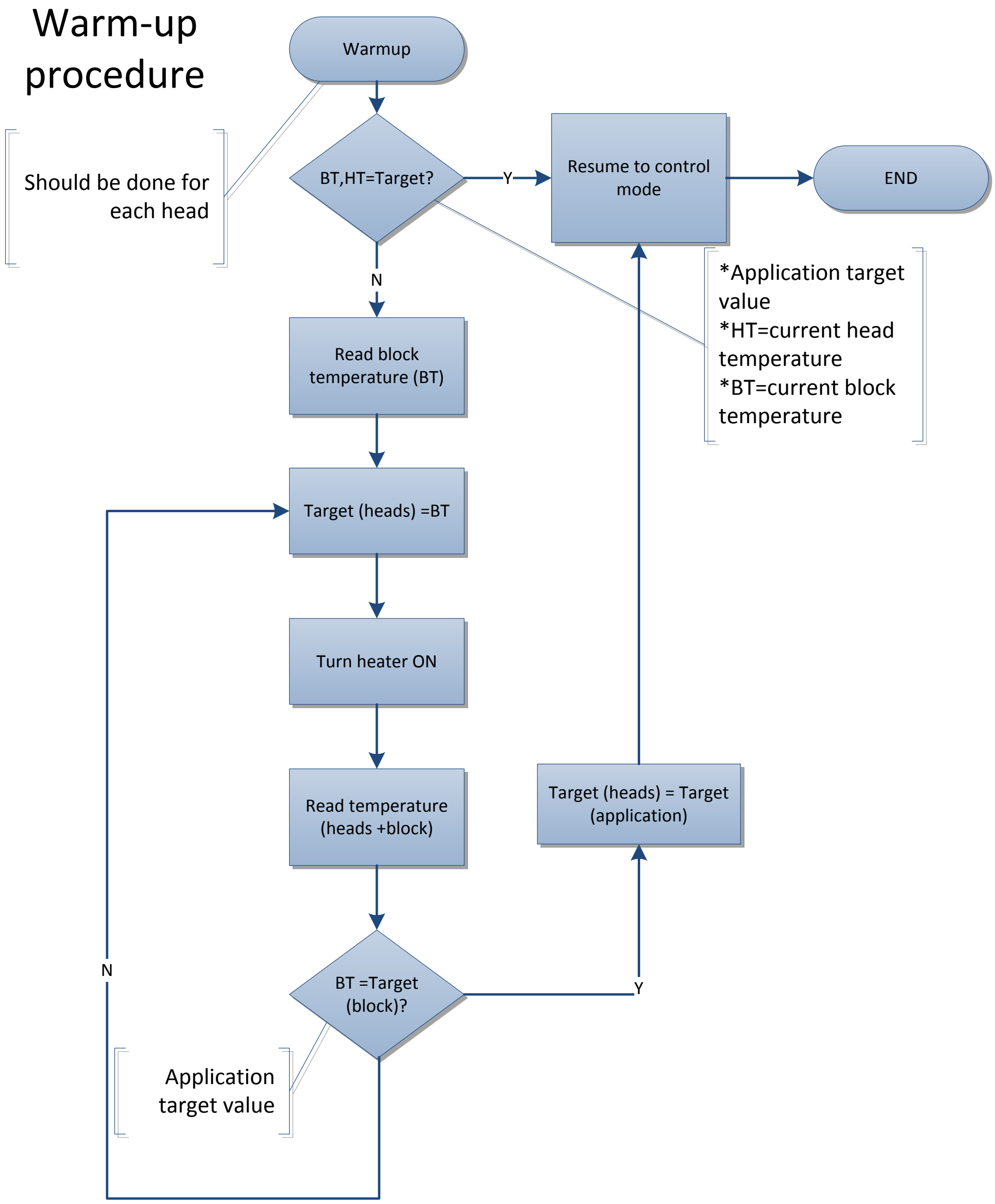
The ***Interval (µs)*** parameter will be entered by user. It defines the time interval between the start of rise of the first pulse and the start of fall of the second pulse.

All the parameters that are described in units of µS, have to be divided by the *Tclock* in order to generate the parameters for the OHDB firmware.

.9. Actuators

.9.1. Material pumps
.9.2. LED illumination
.9.3. Waste pump

Appendix A – warm-up



Appendix B – voltage calibration

- Background

The process (described in the flow chart below) includes the following steps:

.1. Requested-voltage entry
.2. Check if voltage is in range
.3. Calculate potentiometer value for the requested voltage
.4. Calibrate voltage
.5. Save potentiometer value obtained after the calibration in the machine parameters.

- Parameters

Parameter	value
R1 - Model	TBD (M.Levi)
R2- Model	TBD (M.Levi)
Vref	TBD
POT value	1
Min_V	19V
Max_V	33V

- Implementation

.1. formulae

.1.1. Potentiometer- value calculation formula

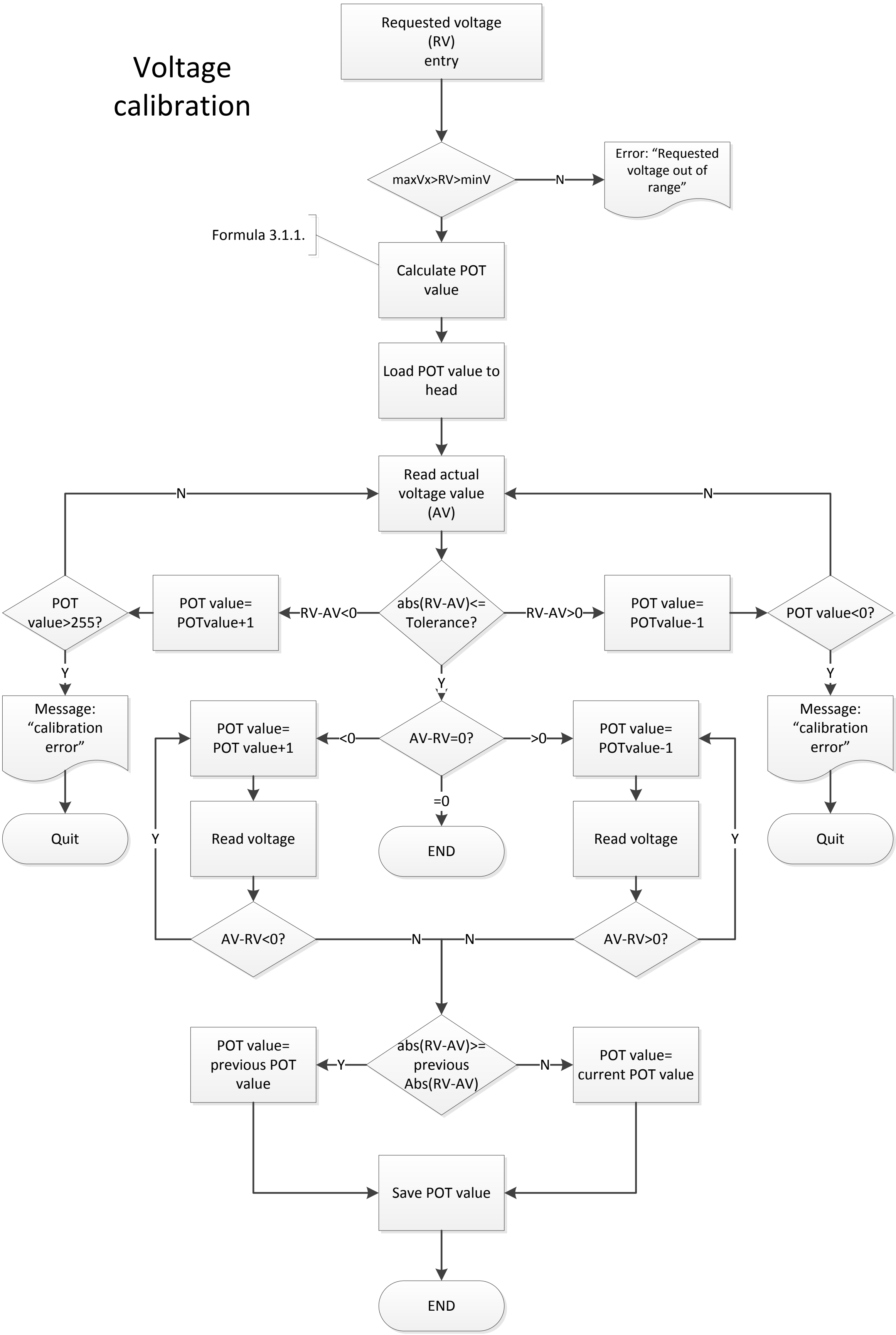
$$N = \frac{(R2+POT\ value)\times(RV-Vref)-R1\times Vref}{RV}\times 256$$

RV – requested voltage

4.2. Calibration process

The process is described in the following flow chart (the process is identical for both support and model heads, except for the parameters).

Voltage calibration



Version control:

Version 01	1-6-2013	draft
Version 02	5-11-2013	Initial version
Version 3	13-11-2013	Auto voltage range setting cancelled
Version 4	18-11-2013	
Version 5	2-2-2014	Warm up procedure changed. Sanity check on frequency. Pulse-builder changed. Strobe duration parameter added.
Version 6	9-2-2014	Conversion formulae changed due to reference change in A/D
Version 7	16-2-2014	Pulse builder changes: • Resonance field removed • Interval field added for double pulse. Will be entered manually