**Voltage calibration and Pulse Delay setting**

**For Objet500 (Dual driver-board support)**

**Software Design Review**

****

**Written by: Guy Ezra**

**May 2014  
 DR Version: 4.0**

Table of Contents

[1. Introduction **2**](#_Toc327971830)

[1.1 Background 3](#_Toc327971831)

[1.2 Design Goals 3](#_Toc327971832)

[1.3 Abbreviations and Acronyms 4](#_Toc327971833)

[2. Architecture and Implementation 4](#_Toc327971834)

[2.1 Time sequence 5](#_Toc327971835)

[2.2 Hardware 6](#_Toc327971835)

[2.3 OHDB Firmware 6](#_Toc327971836)

[2.4 User Interface 7](#_Toc327971837)

[2.5 Embedded Application 8](#_Toc327971838)

[2.6 Parameters 9](#_Toc327971839)

[2.7 Logs 10](#_Toc327971839)

[3. Development 10](#_Toc327971843)

[3.1 Limitations & Risks 10](#_Toc327971844)

[3.2 Development Stages 10](#_Toc327971845)

[4. Operation 10](#_Toc327971846)

[4.1 Limitations 10](#_Toc327971847)

[4.2 Installation 11](#_Toc327971848)

[5. Testing](#_Toc327971849) 11

[6. Effort Estimation **12**](#_Toc327971849)

# Introduction

## Background

Due to printing quality problems in HS and DM modes it was found necessary to change the jetting pulse parameters for these modes. In particular the Pulse Delay needs to be changed from 15 µsec to 18 µsec. With the changed pulse parameters the required head voltages become lower on average by 6V. This poses a problem for Support heads since the required voltages for the majority of the heads becomes lower than the limit provided by the head-drives.

As a result, a new version of head-drives will replace the existing one. The alteration consists of the replacement of 2 resistors, enlarging the allowed voltage range from 19.5-36V to ~15-36V.

The goal of the software change in this PDR is to enable optimal quality printing for any printer that has either new, old or a combination of new and old head-drives:

There will be two sets of Pulse Parameters per head (old vs. new), enabling setting the best parameters for that head.

* Model heads will all be given the set of new parameters.
* Support heads will be given the new set if their head-drive is of the new version and the old set otherwise.

The EM program will identify the head-drive version (according to the manufacture date).

The head calibration process code will use the correct set of resistors for each head (according to the identification of the head-drive version).

The EM will choose the set of pulse parameter to be provided for each head.

## Design Goals

1. To enable accurate head calibration for both the new and the old head drives.
2. To make sure that the code is partible and configurable. It will be easy to reuse the code for other printers.

**Embedded software.**

1. Change communication between application to OHDB.
2. Make the calibration depends on parameters.
3. Possibility to distinguish between old to new HD.
4. Enable Head drive calibration by the correct parameters, the parameters are depends on:
5. - Print mode. (HQ, HS, DM)  
   - Head type (old, new).  
   - Head material (model, support)
6. Determinate the minimum voltage for each head by potentiometer.

## 1.3. Abbreviations and Acronyms

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| HD | Head Driver. |
| EEPROM | Electricity Erasable Programmable Read Only Memory. |
| OHDB | Objet Head driver Board. |
| SW | Software. |
| DR | Design Review. |
| EM | Embedded application. |
| HW | Hardware. |

# Architecture and Implementation

## 2.1. Choosing head type Diagram



## 2.2. Hardware

The old parameters will remain in the same address. (Green)

The new parameters for the new heads will get the new address. (Blue)

**The hardware mapping:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameter** | **Size** |  | **Address** | **Description** |
| PULSE\_WIDTH\_AND\_DELAY\_REG\_ADDRESS | Byte | Low Byte | 0x08 | Pulse width ( old head) |
| PULSE\_WIDTH\_AND\_DELAY\_REG\_ADDRESS | Byte | High Byte | 0x08 | Pulse delay ( old head) |
| POST\_PULSER\_DELAY\_RED\_ADDR | Byte | Byte | 0x51 | Post pulser delay ( old head) |
| HEAD\_1\_DELAY\_REG\_ADDR | Byte | Byte | 0x24 | Head Delay ( old head) |
| PULSE\_WIDTH\_2\_AND\_DELAY\_REG\_2\_ADDRESS | Byte | Low Byte | 0x82 | Pulse width ( new head) |
| PULSE\_WIDTH\_2\_AND\_DELAY\_REG\_2\_ADDRESS | Byte | High Byte | 0x82 | Pulse delay ( new head) |
| POST\_PULSER\_DELAY\_2\_RED\_ADDR | Byte | Byte | 0x83 | Post pulser delay ( new head) |
| HEAD\_2\_DELAY\_REG\_ADDR | Byte | Byte | 0x81 | Head Delay ( new head) |
| HEAD\_DRIVE\_SELECT\_REG\_ADDR | Byte | Low Byte | 0x80 | Each bit represents head. 0 – old value 1- new value Bit number “0” represent head 0. Bit number “1” represent head 1. |

## 2.3. OHDB Firmware

1. Changing OHDB version to “10.5”.
2. Rewrite one communication message between EM to OHDB.

The old message was:  
 struct **TSetPrintingHeadsParamsMsg** {  
 BYTE MessageID;  
 BYTE PulseWidth;  
 BYTE PulseDelay;  
 BYTE Resolution;  
 BYTE PrePulserDelay;  
 BYTE PostPulserDelay;  
 BYTE PrintDirection;  
 BYTE HeadDelayRegisters[8];   
 }STRUCT\_**ATTRIBUTE**;  
 **The new message is:**  
 struct **TSetPrintingHeadsParamsMsg** {  
 BYTE MessageID;  
 BYTE PulseWidth;  
 BYTE PulseDelay;  
 BYTE Resolution;  
 BYTE PrePulserDelay;  
 BYTE PostPulserDelay;  
 BYTE PrintDirection;  
 BYTE HeadDelayRegisters; // Only one byte from the array was used by HW, thus we change the array to one byte  
 BYTE PulseWidth2; //new parameter  
 BYTE PulseDelay2; //new parameter  
 BYTE PostPulserDelay2; //new parameter  
 BYTE HeadDelayRegisters2; //new parameter  
 BYTE HeadTable; //new parameter -each bit represent head. “0” for old head, “1” for new head.   
 }STRUCT\_**ATTRIBUTE**;

1. MessageDecodeOHDB() waits for “TOHDBSetPrintingHeadsParams” message and update the relevant registers as was described in section 2.2 table.
2. **Rewrite the function**:

void MessageDecodeOHDB(BYTE \*Message)

{

case SET\_PRINTING\_HEADS\_PARAMS\_MSG:

{

// Copy the new fields into “TSetPrintingHeadsParamsMsg” struct..

}

}

1. **Rewrite the function:**

void SetPrintingHeadsParamsTask (BYTE Arg)

{

enum{

// adding more 12 enum

// one for each new field,

// one for each “WAIT\_FOR\_new field.

}

// Add more six cases to handle the cases.   
 }

1. Adding five functions, each function write into the relevant register as was described in section 2.2
2. Removing the irrelevant array “HeadDelayRegisters” from the message.
3. Removing all the irrelevant functions for “HeadDelayRegisters”.
4. Removing all the irrelevant cases for “HeadDelayRegisters”.
5. **Rewrite the function:**

void PrintDrv\_FireAllTask (BYTE Arg)

{

enum{

// replacing RESET\_ALL\_1 with RESET\_STATE\_MACHINE\_1

// replacing RESET\_ALL\_2 with RESET\_STATE\_MACHINE\_2

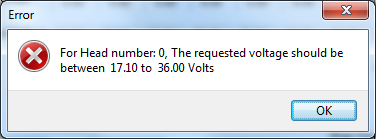
}

// Reset print driver State Mahine instead of Hardware Reset: caused printing with default values

}

## 2.4. User Interface

During calibration, the application finds the minimum voltage for each HD, and check if the requested   
 voltage is between legal limits. If not, message will be display the defect HD and the expected   
 limitation.(Each head has other limitations).



**New requirements:**  
 - Mention the requested volt.  
 - Change the string “Head number 0” to “Head 0”.

If the above message was displayed, the technician needs to replace the head driver.

## 2.5. Embedded Application

**1.** **Add “SetHeadMinVoltage()” function.**  
 The function determines the minimum voltage for each head by:  
 - Initialize the potentiometer array to zero.  
 - Wait for “stabilization time”.  
 - Read the head voltages. (minimum voltage)  
 - Increasing by “0.1” the minimum voltage.  
 - Save each value in parameter manager (“HSW\_MinVoltageArray[i]” parameter)

**2.** **Calling to “SetHeadMinVoltage()” in the following methods:**  
 - CHeadServiceWizard::PrintWTAllModes()  
 - CHeadServiceWizard::PrintCarpetAllModes()  
 - CHeadServiceWizard::SaveAndCalibrate()  
 - CHeadServiceWizard::VoltageAdjustment()  
 - CHeadServiceWizard::FindTargetLayer()  
 - CQuickSHRWizard::AdjustPotentiometerPageEnter()  
 - CQuickSHRWizard::SetHeadsTemperature()  
 - TCalibrationForm::VoltageCalibrateBitBtnClick()

1. **Adding new method** “CHeadHeatersBase::FindResistorValues()“ to distinguish between “new head” to “old head” in order to use “old resistors” or “new resistors”.
2. **Adding new function**

CMachineManager::IsHQMode()  
 {  
 If ((HQ\_INDEX == m\_QualityMode )&&( m\_OperationMode ==SINGLE\_MATERIAL\_OPERATION\_MODE))  
 {  
 return true;  
 }  
 return false;  
 }

**5**. **Calling from** “CPrintControl::SetDefaultConfigParams()” to new method “DetermineHeadParamBeforePrinting()”.  
 The function creates the “TOHDBSetPrintingHeadsParams” massage according to 2.1. diagram.  
 The method finds HD type (Old or New HD), and fill the “HeadTable” table and creating the   
 “TOHDBSetPrintingHeadsParams” message.  
 During preprint Application sends “TOHDBSetPrintingHeadsParams” message to OHDB.

**6.** **Rewrite one communication message between EM to OHDB**.

EM send the “TOHDBSetPrintingHeadsParams” message to OHDB during Preprint.(Please open Section 2.3.3 OHDB

Firmware ). The “TOHDBSetPrintingHeadsParams” struct needs to have the same structure as

“TSetPrintingHeadsParamsMsg” (OHDB code).

**7.** **Change the signature for the following functions (adding head number to the signature**

- CBackEndInterface::CalculatePotValue(int HeadNum, float ReqVoltage);

- CBackEndInterface::EstimateHeadVoltage(int HeadNum, int PotVal);

- CBackEndInterface::CalculatePotValue(int HeadNum, float ReqVoltage);

- CHeadHeatersBase ::CalculatePotValue(int HeadNum, float ReqVoltage);  
 - CHeadHeatersBase ::EstimateHeadVoltage(int HeadNum, int PotVal);  
 - REQUESTED\_VOLTAGE\_LIMITS(\_headNumber, \_minVoltage\_, \_maxVoltage\_)   
 - CHECK\_REQUESTED\_VOLTAGE\_LIMITS(\_val\_, \_excType\_, \_headNumber, \_minVoltage\_, \_maxVoltage\_)

## 2.6. Parameters

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Parameter** | **Default Value** | **Unit** | **Type** | **Hidden** | **File** | **Description** | **Comment** |
| Production\_Year\_For\_New\_Heads | 2015 | Date | int | False | Q2rt.cfg  Q2rt.ref | Production year for the new HD. | The application will read the production year from HD, if the year is lower from **“**Production\_Year\_For\_New\_Heads”, the current HD is old. |
| Resistor**High**Value[2] | Index 0 = 30.1  Index 1 = 12.7 | KOHM | float | False | Q2rt.cfg  Q2rt.ref | The values for **high** resistors. | **Index 0 –** the value for old HD.  **Index 1 –** the value for new HD. |
| Resistor**Low**Value[2] | Index 0 = 1.0  Index 1 = 0.422 | KOHM | float | False | Q2rt.cfg  Q2rt.ref | The values for **low** resistors. | **Index 0 –** the value for old HD.  **Index 1 –** the value for new HD. |
| PulseWidth[2] | Index 0 = 9  Index 1 = 9 | microseconds | float | False | Q2rt.cfg  Q2rt.ref |  | Change the parameter from “float” to “float **array”**.  Index1 will be change in the future.  **Index 0 –** the value for old HD.  **Index 1 –** the value for new HD. |
| PulseDelay[2] | **HS/DM:** Index 0 = 15  Index 1 = 18  **HQ/ Q2rt:**  Index 0 = 30  Index 1 = 30 | microseconds | float | False | Q2rt.cfg  Q2rt.ref  Digital Materials Mode.cfg  Digital Materials Mode.ref  High Speed Mode.cfg  High Speed Mode.ref  High Quality Mode.cfg.cfg  High Quality Mode.cfg.ref  ~Trns commands.cfg  ~Trns commands.ref  ~Trns commands HQ.cfg  ~Trns commands HQ.ref |  | Change the parameter from “float” to “float **array”**.  **Index 0 –** the value for old HD or **support HD**.  **Index 1 –** the value for new HD. |
| PrePulserDelay | 1 | microseconds | int | False | Q2rt.cfg  Q2rt.ref |  |  |
| PostPulserDelay[2] | Index 0 = 25  Index 1 = 25 | microseconds | int | False | Q2rt.cfg  Q2rt.ref |  | Change the parameter from “float” to “float **array”**.  Index1 will be change in the future.  **Index 0 –** the value for old HD.  **Index 1 –** the value for new HD. |
| HeadDelayRegisters[2] | Index 0 = 1  Index 1 = 1 | microseconds | int | False | Q2rt.cfg  Q2rt.ref |  | **Change the array from size 8 to size 2.** (HW uses only one index 4) |
| HSW\_MinVoltageArray[8] | Calibrate during HOW | Volt | float | False | Q2rt.cfg  Q2rt.ref  ~PerMachineDefault.cfg  ~PerMachineDefault.ref  MajorPerMachine.cfg |  | **Change the size array from “4” to “8”.** The values will be change during heads calibration. |
| HSW\_MaxVoltageArray[8] | All values are 36. | Volt | float | False | Q2rt.cfg  Q2rt.ref |  | **Change the size array from “4” to “8”.** |
| RequestedPotentiometerValue | 1.2 | KOHM | float | False | Q2rt.cfg  Q2rt.ref |  | Request potentiometer value |
|  |  |  |  |  |  |  |  |

## 2.7. Logs

1) Main new code for head calibration will be written into the log file under **“HEAD\_HEATERS”** section.  
 The string will start with: **“EEPROM Info:”.**

2) Main new code for printing configuration will be written into the log file under **“LOG\_TAG\_PRINT”** section.  
 The string will start with: **“Print config Info:”.**

**The logs can help to “QA team” testing the implementation.**

\* When identification of a head-drive type is wrong (and therefore the calculation is based on wrong values of the resistors) the calibration process of this head becomes relatively long.

\* The logs do not state which pulse parameters were provided for which head. This can be discovered by sending a Weight-Test. For each head, for the same head-voltage, the weight will be higher if the new set of parameters is used.

For Eden\Connex\Triplex with a 500mm tray the increase in weight will be higher than 0.3gram (much more than the statistical error of Weight-tests).

For HQ – need to check only that the correct resistors are used (the head-drive is identified correctly). There should be no change in the weight-test or any other printing job.

# 3. Development

## 3.1 Limitations & risks

1. During manually “FireAll” sequence, It will determinate “head parameters” the by the last mode printing status (HQ,

HS, DM).

## Development Stages

1. First the development will be done in Objet500 58.11 version.
2. In first stage, we will implement the application side and test it by SIM and Logs.
3. All hardware related functionality will be implemented in an OHDB firmware, It will be test via debugger.
4. Integration with HW (Eldad Hen) - basic tests.
5. Same tests with Meir Bar Natan - running head calibration with different modes, and different HD, running different mode printing.
6. Need to recompile the tester for production department.

# 4. Operation

## Limitations

1. Must use new OHDB card version 10.51 (OHDB\_CPU\_10\_51.hex)
2. Must install new EM application for Objet500 58.1.0.18812 version or higher.
3. Must burn new FPGA. (**aohdb\_54.mcs**)

## Installation

1. Need to recompile the tester for production department.
2. Burn new OHDB firmware (HEX file) version 10.51.
3. Install new embedded executable Objet500 58.11 version
4. Need to burn new FPGA. (HW responsibility)
5. Need to add all the parameters from section 2.6 into “Q2RT.cfg” file.
6. In the future a new version of the calibration excel-file will be used (Dana).

**PLEASE notice, if one of the section is not installed on the machine – It won’t work!!**

# 5. Testing

In the following array[8], you can see eight number, each number is”0” or “1”.  
0 – Represents old HD.  
1 – Represents new HD.  
**On Objet1000: indexes 0-3 are heads models, indexes 4-7 are heads support.**

1. Please check the following files exist after Installation – **You need to burn it:**  
    - “Objet500\ServiceTools\**Cygnal Loader\OHDB\_CPU\_10\_5.hex**” (need to remove the irrelevant .hex)  
    - “Objet500\ServiceTools \Xilinx Burner\aa\_proms\**aohdb\_54.mcs**” (need to remove the irrelevant .hex)
2. Please check the parameters as was described in section 2.6.
3. Manually “fire all”.
4. HD calibration and Weight-Test:

|  |  |  |  |
| --- | --- | --- | --- |
| Mode | HD calibration All heads are new | HD calibration  All heads are old | HD calibration  Mixed old/new # |
| HQ | 11111111 | 00000000 | 11010111 or any other combination for which some Model and Support heads have old parameters and some have the new ones. |
| HS | 11111111 | 00000000 | 11010111 |
| DM | 11111111 | 00000000 | 11010111 |

1. Scanner Transparency:

* Print 3 copies of Scanner Transparency on machine, prior to firmware & software update, scan and analyze with scanner transparency. Save images and XLS of results.
* Print 3 copies of Scanner Transparency on machine, prior to firmware & software update, scan and analyze with scanner transparency. Save images and XLS of results.

**Send all 6 copies of images and XLS results to Shoval.**

5. Test the following Wizards:  
 - Head Optimization vs. Manual calibration (manual weigh-test).

# 6. Effort Estimation

Effort estimation is **14** working days for **Objet500**, including implementation and testing.

|  |  |
| --- | --- |
| Learning the mission | Done |
| Creating DR | Done |
| User Interface. | 0.5 Days - Need to merge from Objet1000 |
| Embedded application 2.5.1- 2.5.3 | 1.5 Days - Need to merge from Objet1000 |
| Embedded application 2.5.3 - 2.5.7 | 3 Days |
| OHDB code changes – sections 2.2 - 2.3 | 3 Days |
| Parameters changes – Section 2.6 | 1 Day |
| Creating new installation | 1 Day |
| SW and HW integration. | 2 Day |
| Basic tests | 2 Days |
| **Total** | **14 Days** |

**\*Time evaluation may be change according to:**

* Machine time availability.
* SRS changes during the development.
* HW readiness. We can’t test it when the HW is not ready on the machine.