**Ambient & Tray Heating**

**Software Design Review**

Written by: Slava Chuhovich

October 2012

Table of Contents

[1. Introduction 3](#_Toc338846152)

[1.1 Background 3](#_Toc338846153)

[1.2 Design Goals 3](#_Toc338846154)

[1.3 Abbreviations and Acronyms 3](#_Toc338846155)

[2. Architecture and Implementation 4](#_Toc338846156)

[2.1 Hardware 4](#_Toc338846157)

[2.2 OCB Firmware 4](#_Toc338846158)

[2.3 User interface 4](#_Toc338846159)

[2.4 Embedded Application 5](#_Toc338846160)

[2.5 Parameters 6](#_Toc338846161)

[2.6 Ambient Heater Activation Logic 7](#_Toc338846162)

[2.7 Abnormals and notifications 9](#_Toc338846163)

[3. Development 9](#_Toc338846164)

[3.1 Limitations & Risks 9](#_Toc338846165)

[3.2 Development Stages 9](#_Toc338846166)

[4. Operation 9](#_Toc338846167)

[4.1 Limitations 9](#_Toc338846168)

[4.2 Installation 9](#_Toc338846169)

[5. Effort Estimation 10](#_Toc338846170)

# Introduction

## Background

The purpose is to prevent the deformations of large models and detachment from the tray during and after the printing, by adding the ambient and tray heating capabilities to the machine. Two ambient heating fans will be added to already existing tray heating mechanism.

## Design Goals

OCB2 firmware

1. Add ambient heaters activation to Actuators unit. (unified and separate activation options)
2. Implement communication task.
3. Add new Embedded – OCB communication message.
4. Add new actuator definitions.

Embedded software

Ambient Heater

1. Add new Embedded – OCB communication message.
2. Add new actuators definitions.
3. Add new actuators to UI at Actuators dialog.
4. Add required parameters to Parameters Manager.
5. Add ambient heater selector enumeration.
6. Implement activation logic.

Tray Heater

1. Split dependency between Tray Heater and Removable Tray functionality.
2. Add required parameters to Parameters Manager.
3. Rename some existing parameters to more appropriate names.
4. Implement runtime object, which is responsible for “tray in place” functionality.
5. Adapt several modules for new functionality.
6. Adapt “python” wizards for new functionality.

## Abbreviations and Acronyms

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| GUI | Graphic User Interface |
| UI | User Interface |
| OCB / OCB2 | Objet Control Board (Main board) |
| PM | Parameter’s Manager |

# Architecture and Implementation

## Hardware

1. Both ambient heaters will be connected to J25, 24V power output on OCB2 card.
2. Tray heater and sensor connections will remain unchanged.
3. The ambient temperature sensing will be performed from OHDB onboard sensor (unchanged).
4. The hardware mapping is as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Actuator name | MCU port | Connector | OCB led | Actuator ID |
| Ambient heater 1 | 5.2 | J25 - 4 | D69 | 10 |
| Ambient heater 2 | 5.5 | J25 - 5 | D82 | 13 |

## OCB Firmware

1. Define new actuator IDs for Ambient Heaters (table above).
2. Implement the activation functionality in the Actuators unit.
3. Implement new OCB – Embedded communication message:  
     
    typedef struct

{

BYTE OnOff;

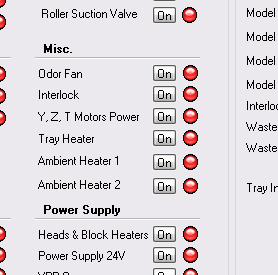
BYTE HeaterSelector;

}TSetAmbientHeaterOnOffMsg;

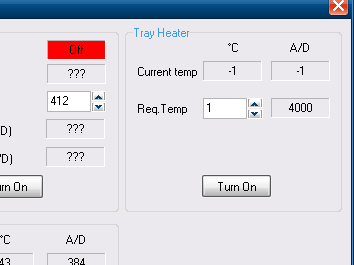
1. Implement SetAmbientHeaterOnOffTask to handle communication.

## User interface

Implement following controls in “Actuators & Sensors” dialog (enabled by AmbientHeaterEnabled param):



Implement following controls in “General Devices” dialog (enabled by TrayHeaterEnabled param):



Show Ambient and Tray temperature indication in main UI:



## Embedded Application

Ambient

1. Implement new OCB – Embedded communication message:  
     
    struct TOCBAmbientHeaterOnOffMessage

{

BYTE MessageID;

BYTE OnOff;

BYTE HeaterSelector;

}STRUCT\_ATTRIBUTE;

1. Implement actuators for each ambient heater. (indexes in table above)
2. Implement activation types enumeration:  
     
    typedef enum

{

AMBIENT\_HEATER\_1 = 1 << 0,

AMBIENT\_HEATER\_2 = 1 << 1

} TAmbientHeaterSelector;

1. Implement CActuator::SetAmbientHeaterOnOff method, which receives the heater ON / OFF state and TAmbientHeaterSelector parameter.
2. Implement CActuator::AmbientHeaterOnOffResponse method to receive ACKs.
3. Expose heaters activation to Roster, to be used in python console or wizards if required.
4. Expose heaters activation to Back-End interface.
5. Implement periodical activation logic in CAmbientTemperature::TemperatureControl method, according to already existing ambient cooling (fan) logic.
6. Writing to application log heaters activation / deactivation.
7. Implement an ability to periodically suspend automatic ambient heaters activation / deactivation.
8. Turn OFF ambient heater, which is in print-block proximity (Heater 1), during Purge sequence.

Tray

1. Remove tray heater and removable tray dependency by splitting it into 2 different modules.
2. Implement CTrayPlacer class.
3. Create “tray placer” runtime objet in Machine Sequencer.
4. Adapt all relevant modules to new implementation.
5. Adapt “python” wizards to new implementation.
6. Add TrayHeaterEnabled parameter.
7. Make tray heating feature an UI TrayHeaterEnabled parameter value dependent.

## Parameters

1. Add AmbientHeaterEnabled check box type parameter. (default value is “checked” on Objet1000 and “unchecked” on other machines)
2. Removed TrayHeaterAndInsertedBypass parameter. Replaced by AmbientHeaterEnabled and RemovableTray parameters.
3. Rename AmbientTemperatureFanControl parameter to WorkingAmbientTemperature.
4. Rename AmbientFanControlByPass parameter to AmbientControlByPass.
5. Rename AmbientTemperatureByPass parameter to AmbientTemperatureErrorByPass.

## Ambient Heater Activation Logic

Following logic applies only if AmbientHeaterEnabled parameter is enabled.

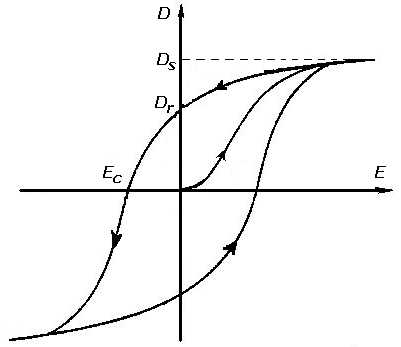
Sequences



Printing

This logic is applied each 10 seconds.



[](http://upload.wikimedia.org/wikipedia/commons/4/4d/Ehysteresis.PNG)

Heating ON

Heating OFF

## Abnormals and notifications

1. If AmbientTemperatureErrorByPass parameter is disabled, following error checks will be performed:
   1. If ambient temperature is under HeadAmbientTemperatureLowLimit parameter value, the printing will be stopped with: “Ambient temperature is under limit (too cold)” message.
   2. If ambient temperature is over HeadAmbientTemperatureHighLimit parameter value, the printing will be stopped with: “Ambient temperature is over limit (too hot)” message.

# Development

## Limitations & Risks

None

## Development Stages

1. All development will be done in Trunk.
2. Ambient Heater feature will be activated by AmbientHeaterEnabled parameter in Parameter’s Manager.
3. All hardware related functionality will be implemented in an OCB2 firmware code.
4. At initial stage, the development and testing will be done using “offline” OCB2 card.
5. At more advances stages of development the testing will be done on an Objet1000 Alpha machine.

# Operation

* AmbientHeaterEnabled parameter must be checked in Parameters Manger in order for ambient heater to work.
* If AmbientControlByPass parameter is checked, ambient fan (cooler) will be permanently ON and ambient heater will be permanently OFF.
* If AmbientTemperatureErrorByPass parameter is checked, ambient errors will be ignored.

## Limitations

* Must use new OCB2 card with Silabs C8051F12x controller.

## Installation

1. Install Ambient Heater device by connecting it’s connector to J25 OCB2 port.
2. Burn new OCB firmware (HEX file).
3. Install new embedded executable.

# Effort Estimation

Effort estimation is 10 working days, including implementation and testing.