***UV Safety Watchdog***

(Immobility detection mechanism)

***Software Design Review***

Written by:

Zohar Shechter

March 2012

[1 Introduction 3](#_Toc319589330)

[1.1 Background 3](#_Toc319589331)

[1.1 Design Goals 4](#_Toc319589332)

[1.2 Abbreviations 4](#_Toc319589333)

[2 Architecture 5](#_Toc319589334)

[2.1 Overview 5](#_Toc319589335)

[2.1.1 New tasks in OCB 5](#_Toc319589336)

[2.2 Project code updates 7](#_Toc319589337)

[2.2.1 Code 7](#_Toc319589338)

[2.2.2 Parameters 7](#_Toc319589339)

[3 Operation 8](#_Toc319589340)

[3.1 Limitations 8](#_Toc319589341)

[3.2 Installation 8](#_Toc319589342)

[4 Development 9](#_Toc319589343)

[4.1 Risks 9](#_Toc319589344)

[4.2 Branching 9](#_Toc319589345)

[4.3 Development Stages: 9](#_Toc319589346)

[4.4 Testing 10](#_Toc319589347)

[4.5 Effort Estimation 11](#_Toc319589348)

# Introduction

## Background

Models which are exposed to direct stationary radiation of UV lamps for a long period of time can heat up to a point they can catch fire.

This had happened before, and the chosen solution to deal with this phenomenon was to monitor the movements of axis X, and once immobility is detected for more than T.B.D seconds, a UV safety watchdog is triggered and shut off the lamps.

This DR document describes the implementation of the aforementioned solution.

\*\* In Office machines, it is possible that the cause for the immobility is that the tray was pulled out during print (can happen due to Y axis movement that can bump into the model). In this case, there’s a popup that asks the user to put the tray back in, and an according message is written to the log. This can help us understand if the implementation of the UV watchdog is useful in those rare cases. \*\*

## 

## Design Goals

1. The main goal is to achieve a solution which will not be dependent on the situation of the EM application. The safety watchdog should be triggered successfully even if the EM application is stuck or from any reason does not command the X axis motor to move.
2. This feature must be as simple as possible, to reduce risks
3. **Bottom line – the safety mechanism should not take unnecessary chances, and should always turn OFF the lamps whenever it suspects that something is wrong in axis X movements and / or bad communication with the OHDB (which reports the axis X positions)**

## Abbreviations

SW – Software

DR – Design Review

DOC – Document

EM – Embedded application

OCB – Objet Control Board

OHDB – Objet Heads Data Board

HW – Hardware

GUI – Graphic User Interface

# Architecture

## Overview

### New tasks in OCB

There are 2 new tasks which run constantly in the OCB.

1. **InquireEncoderDataTask** – a new task that constantly requesting the OHDB to send the OCB the current encoder data. The OHDBMsgDecode task parses the incoming XILINX\_DATA message and records the current encoder data.

2. **ImmobilityDetectionTask** – a new task that constantly monitoring for axis X immobility, based on the encoder data which is received from the OHDB.

**The idea is to split the monitoring from the inquiry so the monitoring will not have to consider / be affected by OHDB communication failures. This way, each task remains simple and the monitoring works efficiently regardless of any other task.**

The following diagrams describe these tasks:

IDLE

YES

UV is ON

Send MSG successfully

Inquiry timer expired

YES

Set Inquiry timer

NO

YES

NO

**Inquire Encoder Data Task**

NO

UV is ON

NO

YES

IDLE

YES

Set immobility timer

UV is ON

UV is ON

NO

YES

Axis x moved

Immobility timer expired

YES

NO

Turn UV OFF

Set trigger flag

Open door

NO

YES

NO

**Immobility Detection Task**

## Project code updates

### Code

The mainly affected modules in the OCB are the following:

* **MsgDecode** – Added 2 tasks for requesting encoder data from the OHDB and monitoring immobility of axis X (as described in previous section).

Also, added a new case in ‘EdenMsgDecode’ that handles a message that the EM sends to reset the immobility trigger, and a new case in ‘OHDBMsgDecode’ that parses the encoder data which is received from the OHDB.

* **UVLamps** – Added a new state ‘UV\_SAFETY\_OFF’ to the UV lamps control task, which we get to it when the immobility trigger is set. This state ignores re-ignition attempts, and we leave it once the EM resets this trigger (happens when EM is restarted).

The modifications in the EM are smaller compared to the OCB, and include small additions to the following:

* **AppParams** – adding new parameter for the immobility timer.
* **MachineSequencer** – Sending immobility trigger reset message to the OCB at power-up.
* **UVLamps** – handling the new immobility error notification received from the OCB.

### Parameters

This feature will use the following new parameters:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Parameter*** | ***Category*** | ***Type*** | ***Default value*** | ***Meaning*** | ***Attributes*** |
| ***MotorsImmobilitySafetyTimeoutSec*** | UV lamps | Integer | 1800 | Define how many seconds the UV lamps are allowed to be turned ON while axis-x is not moving | Hidden |

# Operation

## Limitations

1. None

## Installation

Additions to the installation:

1. Regarding the **EM**
   1. The Q2rt file is updated with a new parameter (along with the REF file)
   2. New EXE file that include all the modifications in the code.
      1. Eden F.O.M: version **XX.3** or above
      2. Connex F.O.M: version **XX.1** or above
      3. Triplex F.O.M: version **XX.1** or above
2. Regarding the **OCB**
   1. New HEX file that include all the modifications which are descried in this document. The new version of the OCB needs to be installed along with the new EM.
   2. OCB versions which will carry this functionality:
      1. Eden F.O.M: version **1.58** or above
      2. Connex F.O.M: version **8.1** or above
      3. Triplex F.O.M: version **20.6** or above

# Development

## Risks

1. If something happens to the Electronics, there is simply nothing that the SW could possibly do to turn OFF the UV.
2. The Triplex OCB project is currently using 99% of the microcontroller’s memory resource. This causes the code to demonstrate unpredicted behavior such as tasks that seem to not work etc. This issue is under investigation and needs to be resolved as part of stabilization efforts.

## Branching

* This feature will be implemented in the following branches:
  + Branches/Slava/Trunk4RAD2007 (for future versions of Eden FOM)
  + Releases\Eden260V\_27.3
  + Releases\Eden350V\_36.3
  + Releases\Eden500V\_50.3
  + Trunk\ConnexFamily (for future versions of Connex FOM)
  + Releases\Connex260\_28.1.X.X
  + Releases\Connex350\_37.1.X.X
  + Releases\Connex500\_57.1.X.X
  + Trunk\TriplexFamily (for future versions of Triplex FOM)

## Development Stages:

All the required modifications can be done instantly on any branch and be merged to other branches, with only small relevant modifications between Eden and Connex FOMs.

At all stages of development, the code is constantly tested on the actual machines (no separation to implementation in emulation phase, and correction cycles on machine phase).

## Testing

The following tests have to be performed during and after implementation:

* + **Basic logics tests**
    1. Check that the immobility trigger is set in the following scenarios:
       1. In cases of UV ignition from printing, from General Devices Dialog, and from the Actuators & Sensors dialog.
       2. After the accurate elapsed time (measure with stop-watch)
       3. The trigger is set successfully and the UV lamps are turned OFF, and the EM shows a proper popup message box that indicates what happened, and what to do.
    2. Check that after the trigger is set, the OCB will always send immobility error notifications to the EM every time there’s an attempt to turn ON the UV (from printing, General devices dialog, or actuators dialog), and the proper popup message is shown.
  + **Abnormal tests**
    1. Check behavior of entire system upon immobility trigger during normal printing. This can be achieved by the following methods:
       1. On Office machines – E260V / C260, it is possible to print with door cheater, and pull the tray out in between slices (dangerous!!, but will also be a good indication…)
       2. Compile a new EXE that does not sends a request for a new slice, and then it will just get stuck while waiting for the HOST to send a new slice right from the beginning of the job.
       3. The abnormal tests can be performed with QA/Alpha members assistance.

The following tests have to be performed after each merge to another branch:

* + 1. Check that the project compiles successfully after the merge
    2. Upgrade the EM and OCB on the each machine and run the basic logics test set as described above.

\* A Simulator for OCB+OHDB can greatly increase tests efficiency (both normal and abnormal) during SW development, as well as QA tests.

## Effort Estimation

Development on E500V – 12 days

Merge to T500 – 4 days

Merge to rest of Eden FOM – 4 days

Merge to Connex FOM – 6 days

Total EE for implementation + required merges is 26 days.