**Safety System**

**Software Design Review**



**Written by: Guy Ezra**

**November 2012  
 DR Version: 7.0**

Table of Contents

[1. Introduction 3](#_Toc327971830)

[1.1 Background 3](#_Toc327971831)

[1.2 Design Goals 3](#_Toc327971832)

[1.3 Abbreviations and Acronyms 3](#_Toc327971833)

[2. Architecture and Implementation 5](#_Toc327971834)

[2.1 Time sequence 5](#_Toc327971835)

[2.2 Hardware 6](#_Toc327971835)

[2.3 OCB Firmware 6](#_Toc327971836)

[2.4 User Interface 7](#_Toc327971837)

[2.5 Embedded Application 9](#_Toc327971838)

[2.6 Parameters 9](#_Toc327971839)

[3. Development 9](#_Toc327971843)

[3.1 Limitations & Risks 9](#_Toc327971844)

[3.2 Development Stages 10](#_Toc327971845)

[4. Operation 10](#_Toc327971846)

[4.1 Limitations 10](#_Toc327971847)

[4.2 Installation 10](#_Toc327971848)

[5. Testing 11](#_Toc327971849)

[6. Effort Estimation 12](#_Toc327971849)

# Introduction

## Background

Safety system mechanism main goal is to protect the user from injures, and to avoid unprotected operations that can harm the printer.

This document defines the required adaptations and the implementation method of the safety system for Objet1000 machine according to the existing mechanism of the interlocks.

Objet1000 has three doors, two emergency stops and one service key.

## Design Goals

1. To avoid motion if one of the doors is open.
2. To avoid motion if one of the emergency stop buttons is pressed.
3. To enable motion in “Technician state”- If one or more doors are opened but the service door is closed and emergency stop buttons are unpressed, the software and the HW enabling motion.
4. To make sure the code is partible and configurable.
5. To make sure the code is easy for making changes for new printer in the future,  
   so It will be easy to add more doors/emergency stop.
6. Display separately the status for each door/emergency stop button in the GUI.

Embedded software.

Door module:

1. Change OCB communication, message structure.
2. Add new sensor definitions for each door.
3. Add a new sensor to UI at Actuators dialog.
4. Change the interface and handle doors array instead of one door.
5. Change in the class “CDoor” all the member functions according to change we made in section 5.

Safety System module.

1. Add new Embedded OCB communication message.
2. Add new actuators definitions.
3. Add new actuators to UI at Actuators dialog.
4. Add required parameter to Parameters Manager.
5. Add new “CSafetySystem” class and a dummy class.

## 1.3 Abbreviations and Acronyms

|  |  |
| --- | --- |
| **Abbreviation** | **Description** |
| GUI | Graphic User Interface |
| UI | User Interface |
| OCB | Objet Control Board (Main board) |
| SW | Software |
| DR | Design Review |
| EM | Embedded application |
| HW | Hardware |

# Architecture and Implementation

## 2.1. Time sequence -

**During Power up:**

Embedded

Initialize the “Normally open” interlock feedback

OCB

Acknowledge message

**When ever trying moving the motor:**

Ask for Safety system status (emergency stop & service key)

OCB

Embedded

The OCB returns current state for Safety System

Ask for doors status

The OCB return the doors status

Are all emergency stop buttons unpressed ?

NO

Yes

Yes

Is service key inserted?

Yes

No

No

Are all other doors closed?

## 2.2. Hardware

Doors devices will be connected to J4 and J5 power outputs on OCB2 card.

Emergency stop devices will be connected to J22 power outputs on OCB2 card.

**Normally open interlock** - The interlock is short circuit

**Normally close interlock -** The interlock close the circuit

Objet1000 interlock/emergency stop were connect by “Normally open” interlocks.

It means if the door is open, the HW signal is “0”

It means if the door is close, the HW signal is “1”

It means if the emergency stop is unpressed, the HW signal is “1”

It means if the emergency stop is pressed, the HW signal is “0”

The doors behavior is inverse compared to all other printers(Normally closed interlock feedback)

The hardware mapping:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Interlock** | **Desc** | **Safety  brd con #** | **ocb brd con #** | **cpu port** | **SensorID** |
| SSR\_FB | Safety switch (emergency stop) rear | J5(1) | J22(1) | P3.0 | 6 |
| SSF\_FB | Safety switch (emergency stop) front | J5(2) | J22(2) | P3.1 | 7 |
| IS\_FB | Interlock service door | J5(3) | J4(1) | P2.4 | 2 |
| IR\_FB | Interlock front door Right | J5(4) | J4(2) | P2.5 | 3 |
| IL\_FB | Interlock front door Left | J5(5) | J5(1) | P2.6 | 4 |
| Service Mode | Service key (interlock bypass) | J5(6) | J5(2) | P2.7 | 5 |

Service Key

Emergency stop front

Emergency stop rear

Right door

service door

Left door

Safety board

## 2.3. OCB Firmware

**The new design is based on the existing design, and the existing code.  
The document focus on the changes we need to make.**

1. Define new sensors IDs for three doors (table above).
2. Change the existing door unit, and fit the structure for three doors.
3. Use the old task (“DoorControlTask”), and split it to three tasks, in order to keep the synchronization between the tasks.
4. Change the communication message structure for OCB – Embedded:  
     
   typedef struct {

BYTE MsgId;

BYTE DoorRequest[NUM\_OF\_DOORS];

BYTE DoorStatus[NUM\_OF\_DOORS];

}TDoorStatusMsg;

1. In “MsgDecode.c”, need to update the task (“GetDoorStatusTask), according to the above structure.

Implement : TaskPtr DoorControlTaskFunctions[] =

{

ServiceDoor\_ControlTask,

FrontRightDoor\_ControlTask,

FrontLeftDoor\_ControlTask,

};.

1. Define new sensor IDs for two emergency stop buttons (table above).
2. Define new sensor IDs for service key (table above).
3. Implement new OCB – safety system communication message:  
     
    typedef struct {

BYTE MsgId;

BYTE EmergencyStopStatus[NUM\_OF\_EMERGENCY\_BUTTONS];  
 BYTE ServiceKeyStatus;  
} TSafetySystemStatusMsg;

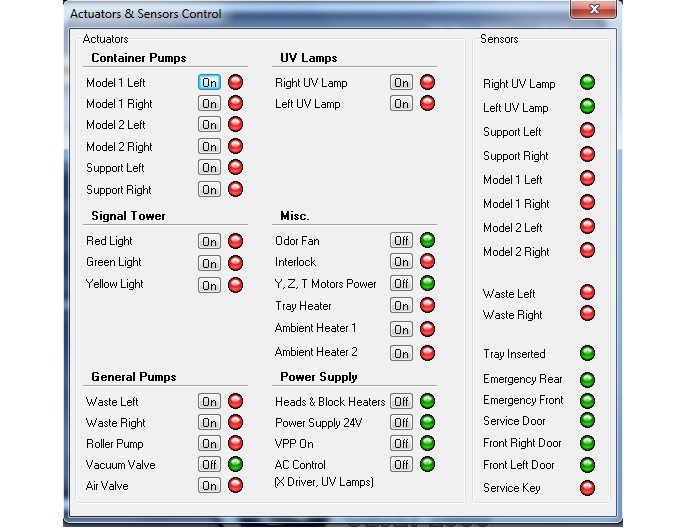
1. Add Task “GetSafetySystemStatusTask”, under “msgDecode.c” . It’ll handle the request for embedded.

## 2.4. User Interface

Implement image for four doors and two for emergency stops. If all the six images are green, the printer is safe to use. If one of the images is red, the printer is not ready for use.

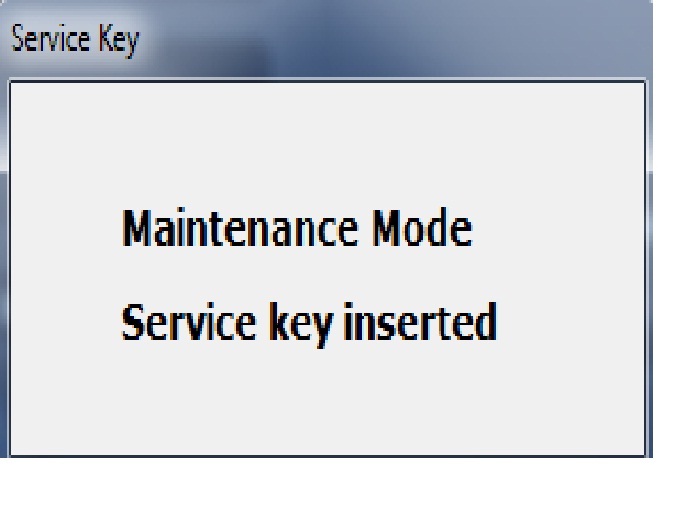
“Actuators & Sensors” dialog:

\



**When trying to move the motor and the operation is not valid, Pop up will show mention the specific problem**, this message box will replace the old message box in triplex.( “Door cover is open”)  
  


**When service key is inserted, The following Pop up message will show**:



## 2.5. Embedded Application

1. Add “CSafetySystem” class to the project.

2. Implement “CSafetySystemBase” and “CSafetySystemDummy” classes to handle “connected” and emulation   
 functionality.

3. Implement new OCB – Embedded communication message:

“ResetNormallyOpenInetLockFeedBack”- send to OCB message to reset the “normally open interlock” parameter.  
 We send this message only one time during power up.

“CSafetySystemBase::GetStatus()” send to OCB message asking the emergency stop status

struct TOCBGenericMessage{  
BYTE MessageID;

} STRUCT\_ATTRIBUTE;

1. Implement “GetStatusAckResponse()”. It waits for OCB respond.

It gets GetStatusAckResponse

struct **TOCBEmergencyStopStatusResponse** {

BYTE MessageID;  
BYTE EmergencyStopStatus[MAX\_NUM\_OF\_EMERGENCY\_STOP\_BUTTONS];  
BYTE ServiceKeyStatus;  
}STRUCT\_ATTRIBUTE;

5. Implement UpdateEmergencyStopButtonStatus() method, which update the emergency stop status.

1. Implement IsAllEmegencyStopButtonsUnpressed () method, which check if all the emergency stop buttons are in “unpressed” state.
2. Implement IsEmegencyStopButtonUnpressed () method, which check the emergency stop status for specific button.
3. CDoor Class: Change the member “bool DoorClosed” to array “**bool** DoorClosed[MAX\_NUM\_OF\_DOORS]”
4. Change all the member functions in “CDoor” class according the changes in section 8.
5. Add a new member function in “CDoor” class:

bool IsDoorClosed (BYTE doorIndex) – it returns the specific door status.

1. OCBStatus- check Safety system status every one second.
2. Add to CBase “m\_SafetySystem” field.

## 2.6. Parametrs

1. “NormallyOpenInterLockFeedback” –Hidden parameter, using for installation.

default value is ‘1’ (true) on Objet1000 and ‘0’ (false) on other machines.  
 If the parameter is “checked”, the safety system mechanism exist.

1. “NormallyOpenInterLockFeedback” –Hidden parameter, using for installation.

default value is ‘1’ (true) on Objet1000 and ‘0’ (false) on other machines.  
 If the parameter is true, we’re using normally open interlock (HW)

# 3. Development

## 3.1 Limitations & risks

If the technician forget the service key inside, the motor will work when the doors are open.

## Development Stages

1. All development will be done in OBJET1000 Trunk.
2. In first stage, we will update the door module and test it.
3. In second stage, we will update the “Safety system” module and test it.
4. All hardware related functionality will be implemented in an OCB2 firmware code.
5. At initial stage, the development and testing will be done using “offline” OCB2 card.
6. Need to make sure the simulator is being updated, and test it.
7. At more advances stages of development the testing will be done on an Objet1000 Alpha machine.

# 4. Operation

* “Safety system Exists” parameter must be checked in Parameters Manger in order for this feature to work.

## 4.1 Limitations

* Must use new OCB2 card with Silabs C8051F12x controller version 30.24 or greater

## 4.2 Installation

1. Install “emergency stop” electric device by connecting it’s connectors to J22 OCB2 ports.
2. Install “door” device by connecting it’s connectors to J4, J5 OCB2 ports.
3. Burn new OCB firmware (HEX file).
4. Install new embedded executable.
5. Need to add “SafetySystemExist=1” to “Q2RT.cfg” file.
6. Need to add “NormallyOpenInterLockFeedback=1” to “Q2RT.cfg” file.

# 5. Testing

**During the development, my tests were made via the Simulator.  
Rules:**  
1. If the service key is inserted and all emergency stop buttons are unpressed– Enable motion.( Technician state)  
2. If all emergency stop buttons are unpressed and service door is closed and front right door is closed and front left door   
 is closed – Enable motion.  
3. All other cases – Disable motion.

The following tests must be done:

1. Try to move the motor, by reproducing the following scenarios:
2. The expected results were described in the following table, please update the ”Actual result”,
3. If the expected result is “Disable”- make sure the Pop up window mention the correct door/emergency stop button.
4. Please try to run print state, and open the door during printer.  
   The HW should stop the printer ( Old implementation, I just want to be sure noting was changed)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test index** | **Service Door** | **Front right door** | **Front left door** | **Service key** | **Emergency stop rear button** | **Emergency stop front button** | **Expected** | **Comment** | **Actual Result** |
| 1. | Closed | Closed | Closed | Inserted | Unpressed | Unpressed | Enable motion | Technician state |  |
| 2. | Open | Closed | Closed | Inserted | Unpressed | Unpressed | Enable motion | Technician state |  |
| 3. | Closed | Open | Closeed | Inserted | Unpressed | Unpressed | Enable motion | Technician state |  |
| 4. | Closed | Closed | Open | Inserted | Unpressed | Unpressed | Enable motion | Technician state |  |
| 5. | Closed | Closed | Closed | Not Inserted | Unpressed | Unpressed | Enable motion |  |  |
| 6. | Closed | Closed | Closed | Inserted | Pressed | Unpressed | Disable motion |  |  |
| 7. | Closed | Closed | Closed | Inserted | Unpressed | Pressed | Disable motion |  |  |
| 8. | Open | Open | Closed | Inserted | Unpressed | Unpressed | Enable motion | Technician state |  |
| 9. | Open | Closed | Open | Inserted | Unpressed | Unpressed | Enable motion | Technician state |  |
| 10 | Open | Closed | Closed | Not Inserted | Unpressed | Unpressed | Disable motion |  |  |
| 11. | Open | Closed | Closed | Inserted | Pressed | Unpressed | Disable motion |  |  |
| 12. | Open | Closed | Closed | Inserted | Unpressed | Pressed | Disable motion |  |  |
| 13. | Closed | Open | Open | Inserted | Unpressed | Unpressed | Enable motion | Technician state |  |
| 14. | Closed | Open | Closed | Not Inserted | Unpressed | Unpressed | Disable motion |  |  |
| 15 | Closed | Open | Closed | Inserted | Pressed | Unpressed | Disable motion |  |  |
| 16. | Closed | Open | Closed | Inserted | Unpressed | Pressed | Disable motion |  |  |
| 17. | Closed | Closed | Open | Not Inserted | Unpressed | Unpressed | Disable motion |  |  |
| 18. | Closed | Closed | Open | Inserted | Pressed | Unpressed | Disable motion |  |  |
| 19. | Closed | Closed | Open | Inserted | Unpressed | Pressed | Disable motion |  |  |
| 20. | Closed | Closed | Closed | Not Inserted | Pressed | Unpressed | Disable motion |  |  |
| 21. | Closed | Closed | Closed | Not Inserted | Unpressed | Pressed | Disable motion |  |  |
| 22. | Closed | Closed | Closed | Inserted | Pressed | Pressed | Disable motion |  |  |
| 23. | Open | Open | Open | Inserted | Unpressed | Unpressed | Enable motion | Technician state |  |
| 24. | Open | Open | Closed | Not Inserted | Unpressed | Unpressed | Disable motion |  |  |
| 25. | Open | Open | Closed | Inserted | Pressed | Unpressed | Disable motion |  |  |
| 26 | Open | Open | Closed | Inserted | Unpressed | Pressed | Disable motion |  |  |
| 27. | Open | Closed | Open | Not Inserted | Unpressed | Unpressed | Disable motion |  |  |
| 28. | Open | Closed | Open | Inserted | Pressed | Unpressed | Disable motion |  |  |
| 29. | Open | Closed | Open | Inserted | Unpressed | Pressed | Disable motion |  |  |
| 30. | Open | Closed | Closed | Not Inserted | Pressed | Unpressed | Disable motion |  |  |
| 31. | Open | Closed | Closed | Not Inserted | Unpressed | Pressed | Disable motion |  |  |
| 32. | Open | Closed | Closed | Inserted | Pressed | Pressed | Disable motion |  |  |
| 33. | Closed | Open | Open | Not Inserted | Unpressed | Unpressed | Disable motion |  |  |
| 34. | Closed | Open | Open | Inserted | Pressed | Unpressed | Disable motion |  |  |
| 35. | Closed | Open | Open | Inserted | Unpressed | Pressed | Disable motion |  |  |
| 36. | Closed | Open | Closed | Not Inserted | Pressed | Unpressed | Disable motion |  |  |
| 37. | Closed | Open | Closed | Not Inserted | Unpressed | Pressed | Disable motion |  |  |
| 38. | Closed | Open | Closed | Inserted | Pressed | Pressed | Disable motion |  |  |
| 39. | Closed | Closed | Open | Not Inserted | Pressed | Unpressed | Disable motion |  |  |
| 40. | Closed | Closed | Open | Not Inserted | Unpressed | Pressed | Disable motion |  |  |
| 41. | Closed | Closed | Open | Inserted | Pressed | Pressed | Disable motion |  |  |
| 42. | Closed | Closed | Closed | Not Inserted | Pressed | Pressed | Disable motion |  |  |
| 43. | Closed | Closed | Open | Not Inserted | Pressed | Pressed | Disable motion |  |  |
| 44. | Open | Closed | Closed | Not Inserted | Pressed | Pressed | Disable motion |  |  |
| 45. | Closed | Open | Closed | Not Inserted | Pressed | Pressed | Disable motion |  |  |
| 46. | Closed | Open | Open | Inserted | Pressed | Pressed | Disable motion |  |  |
| 47. | Open | Closed | Open | Inserted | Pressed | Pressed | Disable motion |  |  |
| 48. | Open | Open | Closed | Inserted | Pressed | Pressed | Disable motion |  |  |
| 49. | Closed | Open | Open | Not Inserted | Unpressed | Pressed | Disable motion |  |  |
| 50 | Open | Closed | Open | Not Inserted | Unpressed | Pressed | Disable motion |  |  |
| 51. | Open | Open | Closed | Not Inserted | Unpressed | Pressed | Disable motion |  |  |
| 52. | Open | Open | Open | Inserted | Unpressed | Pressed | Disable motion |  |  |
| 53. | Closed | Open | Open | Not Inserted | Pressed | Unpressed | Disable motion |  |  |
| 54. | Open | Closed | Open | Not Inserted | Pressed | Unpressed | Disable motion |  |  |
| 55. | Open | Open | Closed | Not Inserted | Pressed | Unpressed | Disable motion |  |  |
| 56. | Open | Open | Open | Inserted | Pressed | Unpressed | Disable motion |  |  |
| 57. | Open | Open | Open | Not Inserted | Unpressed | Unpressed | Disable motion |  |  |
| 58. | Open | Open | Open | Not Inserted | Pressed | Unpressed | Disable motion |  |  |
| 59. | Open | Open | Open | Not Inserted | Unpressed | Pressed | Disable motion |  |  |
| 60. | Open | Open | Open | Inserted | Pressed | Pressed | Disable motion |  |  |
| 61. | Open | Open | Closed | Not Inserted | Pressed | Pressed | Disable motion |  |  |
| 62. | Open | Closed | Open | Not Inserted | Pressed | Pressed | Disable motion |  |  |
| 63. | Closed | Open | Open | Not Inserted | Pressed | Pressed | Disable motion |  |  |
| 64 | Open | Open | Open | Not Inserted | Pressed | Pressed | Disable motion |  |  |

# 6. Effort Estimation

Effort estimation is 23 working days, including implementation and testing.

|  |  |
| --- | --- |
| Learning the OCB, EM, communication, Gui | 4 Days |
| Door OCB development | 3 Days |
| Door EM development | 2 Days |
| Safety System - OCB development | 3 Days |
| Safety System - EM development | 2 Days |
| Gui development(Door + emergency stop + serviceKey) | 3 Days |
| Testing Using the OCB according to section 5 | 2 Day |
| Testing Using the Simulator | 1 Day |
| Testing the code on Objet1000 machine | 3 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.