

Assignment Showmaster DMX

Date : 11dec2017
By : Dick van Kalsbeek, d.v.kalsbeek@roc-teraa.nl
Hours : 24 SBU (clock hours) each team member
Assessment : Team



Read the assignment carefully and ask questions if anything is not clear.

Introduction

In music concerts extensive lighting is used to make the concert more spectacular. These lights are controlled by a system called DMX512. The total show is setup in a so called 'DMX Universe' and can contain a wide variety of devices. The system consists of a 3 or 5 wire multi-drop differential bus in which every device has an address. The system is basically quite simple where there is a maximum of 512 addresses in the bus. The 'location' of the data is the start address of a device, followed by the number of data packages needed for a device. Since the different devices can contain more or less functions, the address-width can differ.

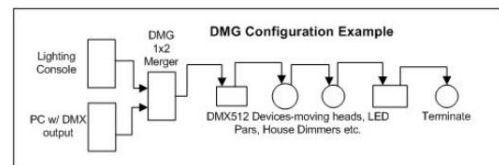
Examples:

- PAR 36 RGB spot, address-width is 6, after the start-address 5 extra addresses must be reserved
- Laser, address-width is 11, after the start-address 10 extra addresses must be reserved



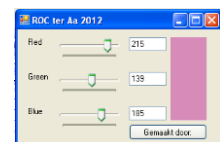
Setup example:

- RIGHT: PAR spot at address 10, Laser at address 16
- RIGHT: PAR spot at address 10, Laser at address 20;
NOTE: 4 addresses unused
- WRONG: PAR spot at address 10, Laser at address 12;
controlling the blue for the spot affects the first address of the laser



In the previous year an application has been built named: ColorMixerDmx. The application used a DLL a hardware layer to the USB/DMX device. The DLL has been replaced by a new version and will be available during the lessons.

Customers tend to draw what they have in mind. They're not supplying you with beautiful wireframes, this is what they want you to do. Below you find a drawing of the customer who wants to have the Showmaster made by you.



IMPORTANT

The assignment requires Object Oriented (OO) techniques. It is probably possible to create an application like this one without using OO but this will deteriorate the maintainability and will increase the development time. OO is an absolute must in this application, creating it without OO techniques will never generate a positive grade.

Method

This assignment is made as a team of 2 team members. A WorkbreakdownStructure(WBS) is needed to examine the required tasks, time and to locate the tasks to the developers. At the assessment the WBS is used as a guideline for more explanation of the executed tasks. So, if a developer was assigned to a task, the knowledge of the task is expected with this developer.

The assignment is described in the projectplan. Also testplan and testresults are available at the assessment date.

Prior knowledge & examples

During the lessons the application is explained. Als the needed DLL and installing is guided. In the previous year a colormixer DMX was built which can be used as a reference. Also the application TuneStore, which uses object arrays has some similarity with ShowMaster.

Requirements (MoSCoW)

In this chapter the requirements are described. This is the way which was used in the first year of the education. This is not the way customer tend to describe they're requirements. In new assignments in the future this must be done by the software engineer.

Must

All musts are required to receive a passing grade (a 5.5). Missing one or more musts immediately results in not passing this assignment.

Should/Could

The shoulds and coulds can increase the grade when all the musts are fulfilled. Normally first the shoulds shall be completed before starting with the coulds, however this assessment does not require this way of working. All shoulds and coulds are rewarded equally.

Won't

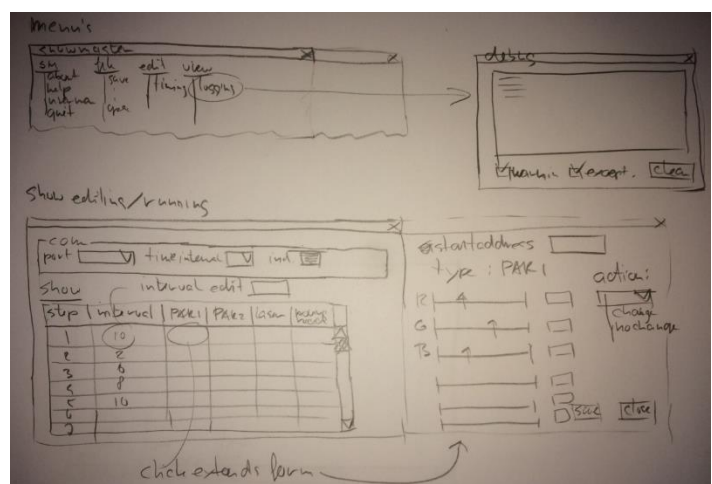
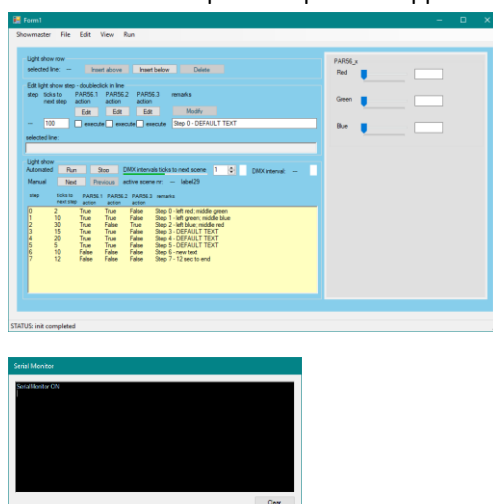
If a won't is not implemented, this will result in an extra point. So, implementing a won't will result in grade decrease.

Tips & resources

- Use classroom examples!
- To be found on N@tschool.
- Make backups!
- Websites references.
- Work together if necessary but beware! Always make sure that you understand what someone did when he helped you. A programmer should at all times be able to explain their own code.

Example wireframes and pictures

Below some examples of a possible application.



Assessment

The assessment is based on the requirements list below. Again, in future assignments the requirements have to be defined by the software engineer.

<div> <div>Showmaster DMX</div> <div>SBU : 24 clock hours created: Dick van Kalsbeek, 23jan2018</div> </div>				name >>	TESTSTUDENT weighing
		Final grade			10.0
		Temp grade in case of lacking musts			10.0
		Progress, enter in report in case of faulty grade			100%
		Class:			
		Date checked:			
		Checked by:			
		Processed:			
parts in application and weighing factors (open group!)					
MoSCoW processing (open group!)					
id	Intro year	task	part in application (weigh factor!)	MoSCoW	
1	2018	application in one form (show control, timer, com port selection)	graphical design	M	1
2	2018	debug logging - separate form	graphical design	M	1
3		debug logging - option to select types of messages	logging	C	1
4	2018	show - overview in RichTextBox, Gridview or comparable	graphical design	C	1
5	2018	application - menu in old windows or Apple style - applicationName > file > edit > view > etc	graphical design	M	1
6	2018	show - timing editing function for each step	operation	M	1
7	2018	show - save and save as in textfile	operation	M	1
8	2018	show - save and save as in XML	operation	C	1
9	2018	show - save and save as in DB	operation	C	1
10	2018	show - new show	operation	M	1
11	2018	debug logging - exceptions automatically send an e-mail to developer	logging	C	1
12		debug logging - save and save as in textfile	logging	S	1
13	2018	device - PAR RGB LED light - class	code	M	1
14	2018	device - moving head - class	code	M	1
15	2018	device - laser - class	code	S	1
16	2018	device - mirror (stardancer) - class	code	S	1
17	2018	device - PAR RGB LED light - class	operation	M	1
18	2018	device - moving head - class	operation	M	1
19	2018	device - laser - class	operation	S	1
20	2018	device - mirror (stardancer) - class	operation	S	1
21	2018	device - functions editing	operation	S	1
22	2018	show editor - selecting grid box of device in RichTextBox shows functions of the selected device	operation	S	1
23	2018	device - default function : address selection	operation	M	1
24	2018	device - address conflicts indicated	operation	M	1
25	2018	device - action when activated - modes: 1.modify setting; 2. no change; 3.off	operation	M	1
26	2018	device - action visible in show control	operation	M	1
27	2018	devices - add new devices in separate editing item	operation	C	1
28	2018	timer - at least 3 timer settings: 100ms; 250ms; 500ms	operation	M	1
29	2018	timer - indicator	graphical design	S	1
30	2018	number of DMX addresses in domain - possibility to limit	operation	C	1
31	2018	timer - automatically inhibit timing conflicts	operation	C	1
32	2018	WorkBreakDown - time estimation; developer timing balancing	documentation	M	1
33	2018	coding standard ROC Ter AA - camel casing/pascal casing/hungarian notation/creator identifier	coding standard	M	1
34	2018	show - status bar for help	logging	S	1
35	2018	programming - devices must be classes	code	W	1
36	2018	directory stucture - separate doc directory for files	documentation	S	1
37	2018	Testplan - according to ROC standard	documentation	M	1
38	2018	Testanalyses - according to testplan	documentation	M	1
39	2018	DLL - DMX DLL available	operation	M	1
40	2017	User manual - available in menu as pdf	documentation	S	1