

# **Party Licht Steuerung – Programmmentwurf für Lichttechniker mit LabView**

## **Studienarbeit**

für die Prüfung zum  
Bachelor of Engineering

im Studiengang TIT08I  
an der Dualen Hochschule Baden-Württemberg Mosbach

von

**Tim Berger**

**17. Juni 2011**

Bearbeitungszeitraum:	6. Theoriephase
Matrikelnummer:	115435
Ausbildungsfirma:	Kurtz Holding GmbH & Co. Beteiligungs KG
Gutachter der DHBW Mosbach:	Prof. Dr. Wolfgang Funk

**Zusammenfassung**

**Abstract**

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>III</b>
<b>Abkürzungsverzeichnis</b>	<b>IV</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Aufgabenstellung . . . . .	1
1.2 Anforderungen . . . . .	1
1.3 Aufbau der Arbeit . . . . .	1
<b>2 LabVIEW als Programmiersprache</b>	<b>3</b>
2.1 Entwurfsmuster - Design Pattern . . . . .	3
2.1.1 Master/Slave . . . . .	3
2.1.2 Zustandsautomat . . . . .	3
2.1.3 Erzeuger Verbraucher Design . . . . .	3
Event Handling . . . . .	3
Error Handling . . . . .	4
<b>3 Programm Analyse</b>	<b>6</b>
3.1 Programmablaufplan . . . . .	6
3.2 Ablaufdiagramm . . . . .	6
3.3 Datenfluss Diagramm . . . . .	6
<b>4 User Interface</b>	<b>6</b>
<b>5 Code Implementierung</b>	<b>6</b>
5.1 Auswahl des Design Pattern . . . . .	6
5.2 Timing . . . . .	6
5.3 Auswahl der Datentypen . . . . .	6
5.4 Init und Shutdown Funktion . . . . .	6
5.5 Aufnahme-Funktion . . . . .	6
5.6 Abspiel-Funktion . . . . .	6
5.7 Stopp-Funktion . . . . .	6
5.8 Speichern und Lade Funktion . . . . .	6
5.9 Fehlerbehandlung . . . . .	6

<b>6 Testen</b>	<b>6</b>
<b>7 Anwendung</b>	<b>6</b>
7.1 Stand-Alone Applikation . . . . .	6
7.2 Installer . . . . .	6
7.3 Webservice . . . . .	6
<b>8 Abschließende Betrachtung</b>	<b>6</b>
8.1 Update . . . . .	6
8.2 Information Hidding . . . . .	6
8.3 Erweiterungen . . . . .	6
<b>Literaturverzeichnis</b>	<b>7</b>
<b>Anhang</b>	<b>8</b>
A.1 Text . . . . .	9
A.2 Text . . . . .	9
A.3 Text . . . . .	9
A.4 Text . . . . .	9
<b>Erklärung</b>	<b>10</b>

# Abbildungsverzeichnis

1	Bedienoberfläche für die Party-Lichtsteuerung . . . . .	2
2	VI Demonstration: Links Frontpanel, Rechts Blockdiagramm . . . . .	4

# **Abkürzungsverzeichnis**

**LabView** LV

**LJ** Light Jockey (dt. Lichttechniker)

**NI** National Instruments

# 1 Einleitung

Diese Studienarbeit dokumentiert den Programmentwurf einer Party-Lichtsteuerung, vom Design der Anwendung über die Implementierung bis hin zur Bereitstellung einer lauffähigen Applikation. Die Programmcode wird mit LabVIEW von National Instruments entwickelt.

## 1.1 Aufgabenstellung

Es ist ein Programm für Lichttechniker zu entwickeln, mit dem eine viel zahl von Scheinwerfer angesteuert werden kann.

## 1.2 Anforderungen

Der Light Jockey (LJ) stellt für verschiedene Lichtkanäle Intensität und Farbe ein. Für eine Gruppe von Lichtkanälen (Set) kann eine Wartezeit, Überblendungszeit, Nachlaufzeit und Name eingestellt werden. Wählt der LJ die Schaltfläche zum aufnehmen, öffnet sich ein Fenster in dem die gewünschten Parameter übergeben werden. Nach der Bestätigung durch ein Klick auf die OK-Schaltfläche wird das erstellte Set hinten an die Queue angefügt.

Hat der Bediener einige Sets angelegt, wird mit der Abspiel-Schaltfläche das aufgenommene Programm durchlaufen. Ein Set das abgespielt wird wartet die angegebene Zeit, dann wird die Farbe bis zur Intensität über die Überblendungszeit hochgefahren. Jetzt beginnt die Nachlaufzeit, ist diese verstrichen wird mit dem nächsten Set aus der Queue fortgefahren. Der Bediener kann jeder Zeit eine abspielende Queue mit der Stopp-Schaltfläche anhalten.

Über die Menüleiste kann der Bediener mit dem Menüpunkt „Datei“ die Queue speichern und laden. In beiden Fällen öffnet sich eine Dialogbox in dem nach Speicher- bzw. Ladepfad gefragt wird.

## 1.3 Aufbau der Arbeit

Bla Bla Bla

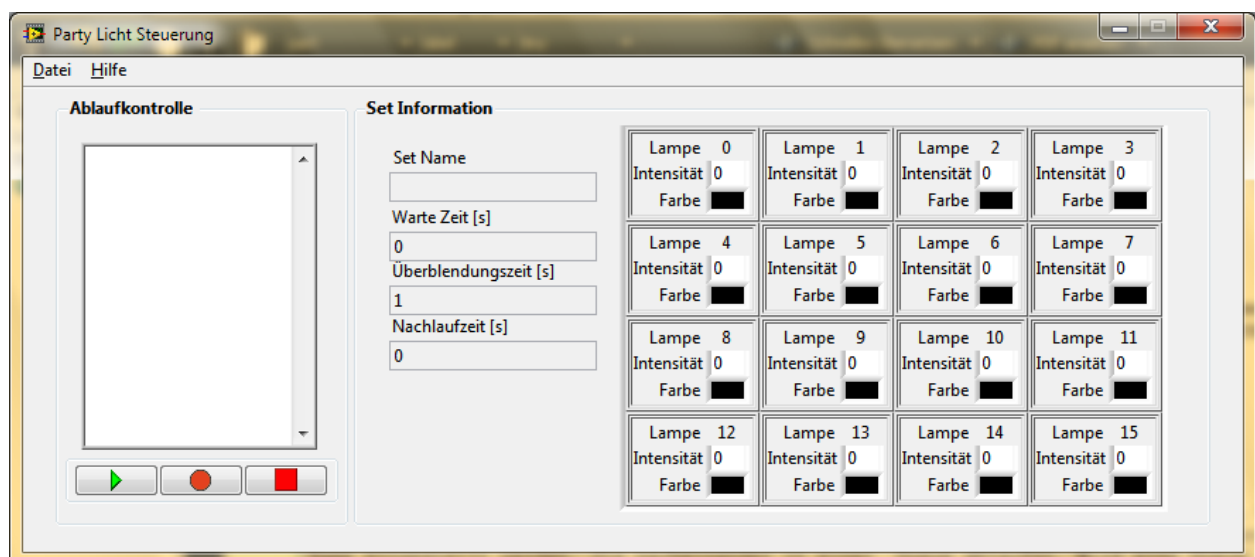


Abbildung 1: Bedienoberfläche für die Party-Lichtsteuerung



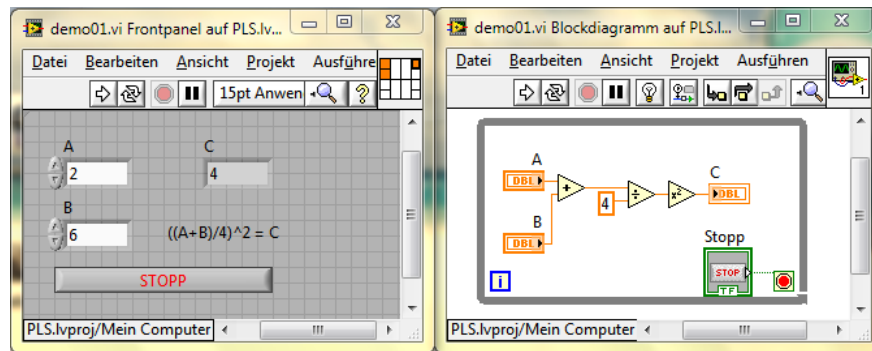


Abbildung 2: VI Demonstration: Links Frontpanel, Rechts Blockdiagramm

## 2 LabVIEW als Programmiersprache

LabVIEW ist ein grafisches Programmiersystem von National Instruments. Das Akronym steht für „Laboratory Virtual Instrumentation Engineering Workbench“.

Die Programmierung erfolgt in der graphischen Programmiersprache „G“. LabVIEW-Programme werden als Virtuelle Instrumente (VIs) bezeichnet. [NI11a]

Sie bestehen aus drei Komponenten:

**Frontpanel** Das User-Interface, über welches der Anwender mit dem VI interagiert.

**Blockdiagramm** Stellt den Programmcode des VIs dar.

**Anschluss** Dient zur Anbindung an weitere VIs.

In LabVIEW liegt die Ausführung von VIs dem Datenflussmodell zugrunde. Ein Blockdiagrammknoten (Bsp. Addition) wird ausgeführt, sobald all seine Eingänge belegt sind. Ist die Ausführung eines Knotens abgeschlossen, werden die Daten an die Ausgabeanschlüsse übergeben und die Ausgabedaten dann an den nächsten Knoten im Datenflussdiagramm weitergeleitet. [?] Die unter LabVIEW erstellten Blockdiagramme werden von einem grafischen Compiler in optimierten Maschinencode übersetzt. Dadurch ist die Performance vergleichbar mit der anderer Hochsprachen wie C oder Pascal. [NI11b]

Abbildung 2 zeigt eine kleine Demonstration. Es wird aus den Eingängen A und B ein Ausgang C berechnet. Die Formel wird im Blockdiagramm abgebildet. Sie lautet:

$$C = \frac{A + B^2}{4}$$

Des Weiteren findet die Berechnung in einer While-Schleife statt. Die Abbruchbedingung ist die Betätigung der Stopp-Schaltfläche.

## **2.1 Entwurfsmuster - Design Pattern**

### **2.1.1 Master/Slave**

### **2.1.2 Zustandsautomat**

### **2.1.3 Erzeuger Verbraucher Design**

### **Event Handling**

### **Error Handling**



## **3 Programm Analyse**

### **3.1 Programmablaufplan**

### **3.2 Ablaufdiagramm**

### **3.3 Datenfluss Diagramm**

## **4 User Interface**

## **5 Code Implementierung**

### **5.1 Auswahl des Design Pattern**

### **5.2 Timing**

### **5.3 Auswahl der Datentypen**

### **5.4 Init und Shutdown Funktion**

### **5.5 Aufnahme-Funktion**

### **5.6 Abspiel-Funktion**

### **5.7 Stopp-Funktion**

### **5.8 Speichern und Lade Funktion**

### **5.9 Fehlerbehandlung**

## **6 Testen**

## **7 Anwendung**

### **7.1 Stand-Alone Applikation**

### **7.2 Installer**

### **7.3 Webservice**

## **8 Abschließende Betrachtung**

## Literatur

[NI11a] NI: *Einführung in LabVIEW - Dreistündiger Einführungskurs*. [http://www.ni.com/pdf/academic/d/labview\\_3\\_hrs.pdf](http://www.ni.com/pdf/academic/d/labview_3_hrs.pdf), 2011. Zugriff am 2.6.2011 um 14:27 in Datei „*labview3hrs.pdf*“.

[NI11b] NI: *Wie funktioniert der Compiler von NI LabVIEW?* <http://zone.ni.com/devzone/cda/tut/p/id/11936>, 2011. Zugriff am 2.6.2011 um 14:27 in Datei „*Compiler LabVIEW - Developer Zone - National Instruments.pdf*“.

# Anhang

<b>A.1 Text</b>	<b>9</b>
<b>A.2 Text</b>	<b>9</b>
<b>A.3 Text</b>	<b>9</b>
<b>A.4 Text</b>	<b>9</b>

**A.1 Text**

**A.2 Text**

**A.3 Text**

**A.4 Text**

## **Erklärung**

Hiermit erkläre ich, dass ich die vorliegende Arbeit mit dem Titel

### **Party Licht Steuerung – Programmentwurf für Lichttechniker mit Lab-View**

selbständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe.

Mosbach, den 17. Juni 2011

Tim Berger