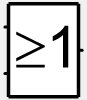
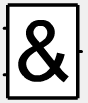
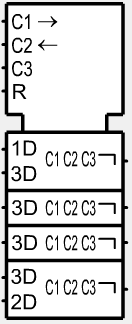
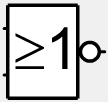
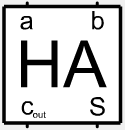
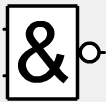
# Logik Simulator



Projektteam: Moritz Wahlenmeier,   
Bastian Schäfer &   
. Stephen Heisser

Auftraggeber: Markus Greiner



Contents

* [Logik Simulator 1](#_Toc10044957)
* [Allgemeine Informationen 3](#_Toc10044958)
* [Projektbeschreibung 4](#_Toc10044959)
* [Funktionsweise 5](#_Toc10044960)
* [Optimierungsmöglichkeiten 6](#_Toc10044961)
* [Zustand 7](#_Toc10044962)
* [Wie es vorgefunden wurde: 7](#_Toc10044963)
* [Geplanter Endzustand: 7](#_Toc10044964)
* [Letztendlicher Endzustand 8](#_Toc10044965)
* [Zeitplan 9](#_Toc10044966)
* [Geplantes Material 9](#_Toc10044967)

# Allgemeine Informationen

**Team:**

Moritz Wahlenmeier   
 moritz.wahlenmeier@yahoo.de

Bastian Schäfer  
 schaefer2001@web.de

Stephen Heisser  
 st.heisser@gmail.com

**Auftraggeber:**

Herr Markus Greiner  
greiner@gds2.de  
Lehrer Gottlieb Daimler Schule 2  
Böblinger Str. 73, 71065 Sindelfingen

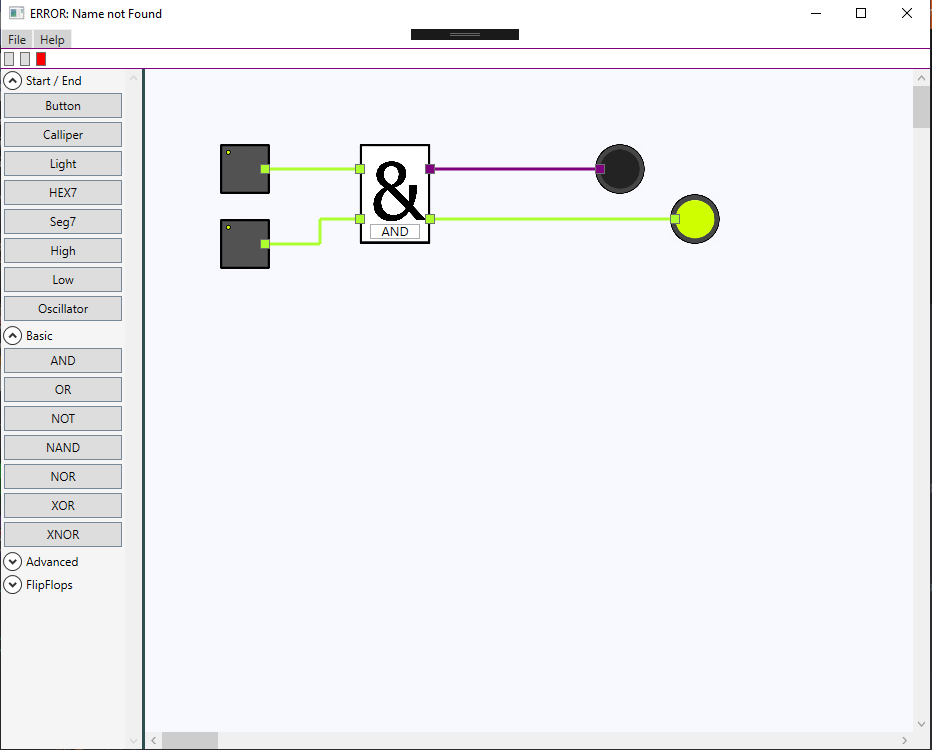
# Projektbeschreibung

Das Ziel ist das ablösen des alten, auf Flash basierendem, Programms „Logiflash“, durch eine auf Windows Presentation Foundation (WPF) basierende Desktopanwendung.   
Da „Logiflash“ alte Symbole beinhaltet und in vielen Situationen nicht mehr von Browsern unterstützt wird. Des Weiteren ist das Programm aktuell nicht sehr benutzerfreundlich. „Logiflash“ wird aktuell im Praktikum Informatik Unterricht im ersten Schuljahr des 2BKI benutzt und soll gegebenenfalls durch das neue Programm abgelöst werden.

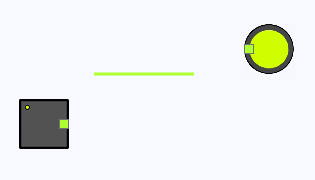
Hierzu muss das Simulationsprogramms zum Erstellen und Testen von logischen Schaltungen erstellt werden.   
Dieses Programm soll einige Optimierungen enthalten, die auf der nächsten Seite gelistet sind (Änderungen können je nach Situation anfallen). So zum Beispiel, dass das Programm auf die gesamte Bildschirmflache skalierbar ist.

# Funktionsweise

In der unteren rechten Leiste befindet sich eine Liste mit allen Logik Gattern als auch Anfangs- und Endbausteine wie Buttons und Lampen, welche per Drag und Drop auf die Arbeitsfläche gezogen werden können. Um diese zu verbinden klickt man auf die Output Fläche (Quadrate rechts an Baustein) und zieht ein Kabel auf die Input Fläche (Quadrate links an Baustein). Kabel können nur vertikal oder horizontal gezogen werden. Auf jedes vertikale folgt ein horizontales Kabel und andersrum. Schaltungen können unter dem Menüpunkt File oder durch Shortcuts in der zweiten Zeile gespeichert, geöffnet oder gedruckt werden. Durch drücken des roten Icons in der zweiten Zeile wird der Lösch-Modus aktiviert oder deaktiviert. Im Lösch-Modus werden Gatter und Kabel durch Anklicken gelöscht.

  
logisch 0: lila logisch1: grün

# Optimierungsmöglichkeiten

****

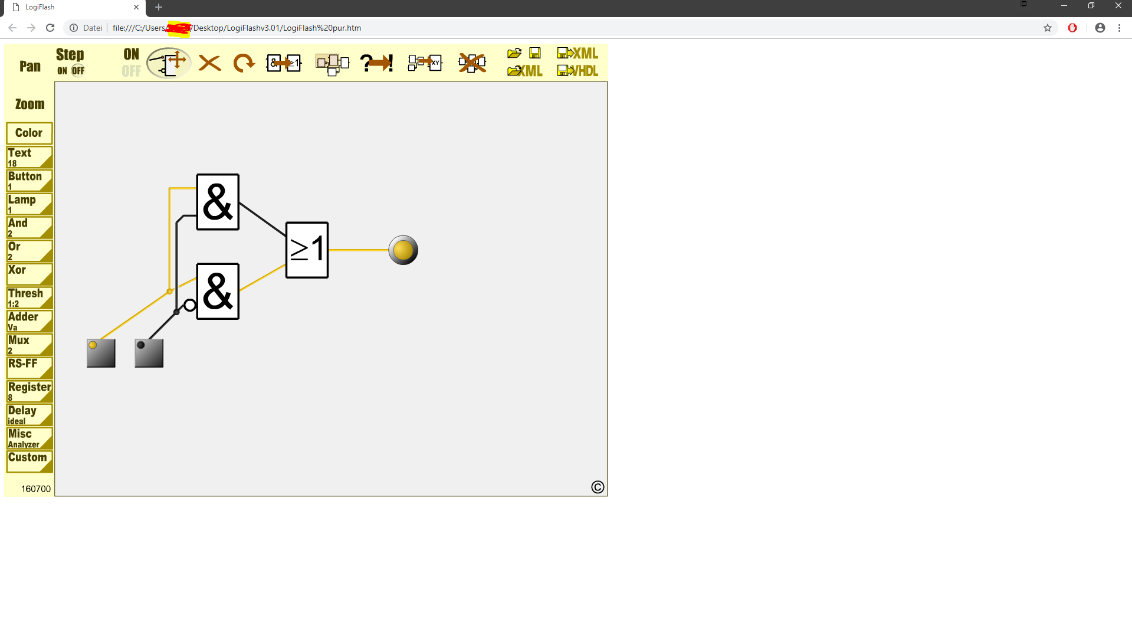
Beim Verschieben der Bausteine bleiben verbundene Kabel an ihrer vorherigen Position die logische Verbindung bleibt aber weiterhin bestehen.

Optimiert: Algorithmus setzt Kabelende /Anfang an neue Position und baut je nach Verschiebung benötigte vertikale/horizontale Kabel ein (Aufwendig).

Es fehlen noch einige Bilder für die FF und komplexeren Schaltungen, die noch angefertigt werden müssen und manche bereits existierende Bilder müssen noch angepasst werden.

# Zustand

## Wie es vorgefunden wurde:



Button2

Button1

* Nicht skalierend abhängig von Bildschirmgröße
* Linien ziehen nicht anwenderfreundlich
* Testfunktion änderbar im laufenden Betrieb
* Flash basierend (wird häufig nicht mehr von Browsern unterstützt)
* Veraltete Symbole
* Umständliches umstellen für Löschen von Bausteinen
* Text ist nicht an Bausteine gebunden und keine Pflicht
* Benötigt ein weiteres Programm, um Schaltungen auszudrucken

## Geplanter Endzustand:

* Skalierend
* auf (WPF) basierende Desktopanwendung
* //Debugging Umgebung
* Tasten-kürzel
* //Möglichkeit simple Programme auf einem Mikrokontroller zu testen
* //Kontextmenü
* Direktes Drucken aus dem Programm heraus
* Feste namens Zuordnung zu Bauteilen

## Letztendlicher Endzustand

# Zeitplan

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Wahlenmeier** | Zeit in h | **Schäfer** | Zeit in h | **Heisser** | Zeit in h |
| Programm Organisation | 5 | Grafikdesign | 20 | Logik Bausteine in C# | 29/10 |
| Programm in C für Mikrokontroller | 15 | Tastatur Kürzel | 10 | Verbindungslinien | 32/15 |
| Schnittstelle für Mikrokotroller zu C# | 10 | Englische und deutsche Bedienelemente | 5 | Abspeichern und öffnen von Programm | 28/15 |
| Drag & Drop  / Toolbox | 10 |  |  | Fleißarbeit | 35/0 |
| Fehlerdiagnose | 15 | Fehlerdiagnose | 15 | Fehlerdiagnose/Cleanup | 53/15 |
| Lern Zeit | 30 | Lern Zeit | 30 | Lern Zeit | 54/30 |
| PowerPoint | 2 | PowerPoint | 2 | PowerPoint | 0/2 |
|  |  |  |  |  |  |
| Gesamtzeit | 87 | Gesamtzeit | 82 | Gesamtzeit | 231/87 |

# Geplantes Material

|  |  |  |  |
| --- | --- | --- | --- |
| **Materialname** | **Anzahl** | **Preis in Euro €** | **Gestellt**  **von Team** |
| XC886 Mikrokontroller | 1 | 60 | X |
| USB-A zu USB-B Kabel | 1 | 1 | X |
|  |  |  |  |
| Gesamtpreis |  | 61 |  |