Hochschule Darmstadt
Fachbereich Informatik

Chaos und Fraktale

Praktikum

Praktikumsaufgabe 1

Semester:	SoSe 2017
Laboranten:	Ken Hasenbank Artur Schmidt
Datum:	03.11.2017

Inhaltsverzeichnis

In diesem Laborversuch soll ein System auf dessen Echtzeiteigenschaften Untersucht werden. Hierzu werden mittels des Messtools von Keil µVision die Ausführungszeiten gemessen und daraufhin überprüft, ob sie kleiner als die obere Schranke von 400 µs sind. Als Anwendung dient hierbei die in der Laboraufgabe 1 entwickelte Fahrstuhlsteuerung.

1 Einleitung

In diesem Laborversuch soll ein Nachweise echter Echtzeiteigenschaften erfolgen. Dies ist deshalb so wichtig, weil es bei eingebetteten Systemen nicht ohne weiteres Möglich ist neue Software aufzuspielen und so nachträglich Softwarefehler zu beheben. Weiterhin macht die in der Regel hohe Stückzahl ein nachträgliches Austauschen der

2 Codeanalyse

3 Testfolge

4 Auswertung

5 Anhang

5.1 Code

Listing 1: Quellcode

```
#include <REG515C.H>

// Laboraufgabe 2
// Tobias Wenzig, Marteyn Weidenbach, Artur Schmidt

// Eingaenge
sbit wunsch1_i = P1^1;
sbit wunsch2_i = P1^2;
sbit in2_i = P1^7;
bit wunsch1;
bit wunsch2;
bit in2;

// Ausgaenge
sbit ab_o = P5^1;
sbit auf_o = P5^2;
bit ab;
bit auf;

// functionen
static void readInputVars();
static void computeOutputVars();
static void writeOutputVars();

typedef enum {keinW, wOben, wUnten} zustaende_t;
zustaende_t zustand = keinW;

void main(){
        while(1){ // Endlosschleife
                // Eingabephase
                readInputVars();
                // Berechnungsphase
                computeOutputVars();
                // Ausgabephase
                writeOutputVars();
        }
}
```

```
static void readInputVars(){
        wunsch1 = wunsch1_i;
        wunsch2 = wunsch2_i;
        in2 = in2_i;
        return;
}

static void computeOutputVars(){
        // Zustandsuebergangsfunktion
        switch(zustand) {
                case keinW:
                        if(wunsch2 && !in2){
                                zustand = wOben;
                        }
                        if(wunsch1 && in2){
                                zustand = wUnten;
                        }
                        break;
                case wOben:
                        if(in2)
                        {
                            if(!wunsch1){
                                zustand = keinW;
                            }
                            else{
                                zustand = wUnten;
                                }
                        }
                        break;
                case wUnten:
                        if(!in2)
                        {
                            if(!wunsch2){
                                zustand = keinW;
                            else{
                                zustand = wOben;
                            }
                        }
                        break;
        }
        // Ausgabefunktion
        switch(zustand){
                case keinW:
                        ab = 0;
                        auf = 0;
                        break;
                case wOben:
                        ab = 0;
                        auf = 1;
```

```
                    break;
                case wUnten:
                    ab = 1;
                    auf = 0;
                    break;
        }
        return;
}

static void writeOutputVars(){
        ab_o = !ab;
        auf_o = !auf;
        return;
}
```

5.2 Zustandsautomat

5.2.1 Testfolge für die vollständige Zustandsübergangsabdeckung

HIER DIE TABELLE mit dem Label Zustand EINFÜGEN! DANKE!