



[Datum]

Widerstandsrechner

Projekt ITA

Erstellt: Bielefeld

Tim Kirchner, Jens Höcker

Inhaltsverzeichnis

0	Vorschlagsphase	1
1	Planungsphase	1
1.1	Arbeits- und Zeitplan	1
1.2	Ist-Beschreibung	1
1.3	Pflichtenheft	1
2	Konzeptphase	2
2.1	Überblick	2
2.2	Grobentwurf der Dokumentation	2
2.2.1	Komplett	2
2.2.2	Planungsorientiert	2
2.2.3	Produktionsorientiert	2
2.3	Grobentwurf des Produktes	2
2.4	Vergleich und Votum	3
3	Entwurf	3
3.1	Benutzer Interface	3
3.2	Datenbeschreibung	3
3.3	Systemaufbau	4
3.3.1	Aufrufdiagramm der Grundversion	4
3.3.2	Aufrufdiagramm der Vollversion	4
3.4	Definition der Module	4
3.4.1	Eingabe der Widerstandsfarbringe	5
3.4.2	Verarbeitung der Eingabe	5
3.4.3	Ausgabe des Widerstandswertes	6
4	Implementation + Test	6
4.1	Implementation	6
4.2	Weitere Software	6
4.3	Hauptprogramme:	6

4.4	Testprogramme:	6
4.5	Nachweis der Funktionalität	7
4.5.1	Disclaimer	7
5	Einführung	7
5.1	Lieferumfang	7
6	Persönliche Erklärung	8
7	Abbildungsverzeichnis	9
8	Tabellenverzeichnis	9
9	Quellenverzeichnis	9
10	Softwareverzeichnis	9
10.1	IDEs	9
10.2	Dokumentation	9
10.3	Datenverwaltung	10

0 Vorschlagsphase

Die Ergebnisse der Vorschlagsphase sind an den Punkten des Dokumentes C73A (Anhang) orientiert.

Es soll ein Widerstandsrechner als Konsolenanwendung erstellt werden. Die Realisierung soll mit vollem Dokumentationsumfang und in Gruppen-/ Partnerarbeit entstehen.

Das Projektergebnis soll eine vollständige Dokumentation und das Produkt selbst enthalten.

1 Planungsphase

1.1 Arbeits- und Zeitplan

Die Arbeitsplanung orientiert sich, ebenso wie diese Dokumentation, an der Projektplanungsvorlage gemäß O31F.

Der geplante Zeitumfang liegt bei vier Zeiteinheiten zu je 90 Minuten. Die Dokumentation als Gesamtergebnis wird über die gesamte Dauer des Projektes erstellt.

Die Realisierung des eigentlichen Produktes erfolgt ab der dritten Arbeitsphase in Anlehnung an den Ablauf einer Projektrealisierung angelehnt. Die Erfüllung der Kann-Kriterien ist als zusätzliche Arbeitsleistung geplant.

1.2 Ist-Beschreibung

Die aktuelle Bestimmung eines Widerstandwertes ist die Berechnung von Hand mithilfe einer Tabelle. Das Berechnungen von Hand ist aufwendig und soll durch das Produkt ersetzt werden. Der hierfür notwendige Datenstamm ist einem Tabellenbuch (vgl. Literaturverzeichnis) entnommen.

1.3 Pflichtenheft

Das Pflichtenheft ist an Abschnitt zwei (Anhang: C73A) angelehnt. Im folgenden die Zielfestschreibung als Muss-Kriterien:

- Die Implementation des Produktes (Software) ist in ANSI-C zu realisieren
- Eingabe: Eine Zeichenkette für einen Widerstandswert. Die einzelnen Farben sollen als Worte repräsentiert und durch „-“ getrennt werden.
- Die genannte Zeichenkette ist als gesamte Eingabe zu verarbeiten
- Das Phasenmodell soll als Planungsgrundlage verwendet werden
- Realisierung des Projektes unter Anwendung des Top-Down Modells
- Zur Implementation ist die C-Standard Bibliothek zu verwenden
- I18N und L10N ist nach Möglichkeit anzuwenden (Internationalisierung und Lokalisierung)
- Die Verwendung von globalen Variablen und Sprüngen ist zu vermeiden

Konzeptphase

- Zu jeder Funktion (der Hilfestellung C73A 3) der Implementation soll eine Testsoftware existieren

Folgend die Auflistung der Kann-Kriterien:

- Als Eingabe sind auch die Kürzel gem. DIN 47002 oder auch IEC 60757 möglich
- Groß und Kleinschreibung sollten egal sein
- Unterscheidung zwischen Kohle-, Metallschicht und Präzisionswiderständen
- Die Realisierung einer grafischen Oberfläche ist vorerst nicht angedacht
- Zusätzliche Erweiterung von Funktionalitäten die praktisch sein können

2 Konzeptphase

2.1 Überblick

Die Realisierung des geforderten Produktes (Software) muss gemäß Pflichtenheft in C erfolgen. Des Weiteren ist neben dem Kernprodukt auch Testsoftware gefordert. Die einzelnen Produktbestandteile werden jeweils entwickelt, implementiert und anschließend getestet.

2.2 Grobentwurf der Dokumentation

2.2.1 Komplet

Eine Dokumentation könnte im gesamten Umfang erfolgen. Dies bedeutet, dass sämtliche Funktionen Datenmodelle und Programmstrukturen sowie Abläufe einzeln entworfen, geplant, dokumentiert und letztendlich implementiert werden. Dies gilt für alle Bestandteile des eigentlichen Produktes sowie der geforderten Testsoftware.

2.2.2 Planungsorientiert

Die Realisierung erfolgt anhand von Dokumentation. Die Dokumentation sollte den Gesamtablauf des Programmes beschreiben. Des Weiteren sind die einzelnen Bestandteile (Funktionen) in ihren Schnittstellen vorgegeben und das Zusammenspiel modelliert. Die Gesamtfunktion ist in einem Ablaufdiagramm (Anhang: Grundaufbau.pdf) geplant und dokumentiert.

2.2.3 Produktionsorientiert

Die Dokumentation wird soweit beschränkt wie es nötig ist. Die Detailstruktur entsteht bei der Realisierung.

2.3 Grobentwurf des Produktes

Das Produkt soll sich an das Pflichtenheft halten. Die Erfüllung der Kann-Kriterien ist angestrebt und von der Realisierbarkeit der vorgegebenen Zeitplanung abhängig.

Es werden zwei Programmversionen angelegt. Die Grundversion erfüllt die Grundanforderungen und die Berücksichtigung von Kürzeln. Die Vollversion berücksichtigt auch Metallschichtwiderstände sowie Präzisionswiderstände. Des weiteren enthält die Vollversion sonstige Erfüllungen von Kann-

Entwurf

Kriterien und vorher nicht definierten Anforderungen wie einer Hilfe-Funktion und dem berechnen des Wertebereiches eines Widerstandes.

2.4 Vergleich und Votum

Der Vergleich sowie eine interne Abstimmung hat ergeben, dass in Bezug auf die geforderte Komplexität des Produktes (Software) und dem gegebenen Zeitrahmen eine (unter 2.2.2 genannte) planungsorientierte Dokumentation der Planungsphase am effizientesten ist. Die Implementation soll im genannten Umfang zum vereinbarten Produkt führen. Da nur die Grundversion die Funktionsaufrufe des Pflichtenheftes benutzt ist diese als Dokumentationskern herangezogen.

3 Entwurf

3.1 Benutzer Interface

Das Produkt ist als Konsolenanwendung definiert. Somit besteht das Benutzerinterface aus einer Konsolenanwendung die mit Dialogen die Interaktion ermöglicht.

Gemäß dem Pflichtenheft kann der Benutzer, nach Aufforderung, die Farben des Widerstandes eingeben. Die Eingabe soll als Einzelne Farben (Ausgeschrieben oder nach Kann-Kriterium als Kurzform) jeweils durch Bindestrich getrennt erfolgen. Es ist (gemäß Kann-Kriterium) die Eingabe von Widerständen mit 4 bis 6 Widerstandsringen möglich. Der Benutzer bekommt anschließend den Widerstandswert in geeigneter Weise zurück. Das gewählte Format ist definiert als 1-999 (Exponenten Kennzeichen k/M/G...) Ohm \pm 1-99% Toleranz mit \pm Temperaturkoeffizient in ppm.

3.2 Datenbeschreibung

Die Datenstrukturen sind gemäß MWC-Prinzip getrennt von der Implementation der Programmfunktionen zu betrachten.

Für die Eingabe wird ein char [] (im folgenden String) genutzt.

Innerhalb der Software wird kein digitales Abbild des Widerstandes benutzt. Die Benutzereingabe bleibt erhalten und dient als Datenstamm. Die Benutzereingabe wird im gesamten Verlauf (nach der Anpassung auf Kleinbuchstaben) nicht mehr geändert.

Die Ausgabe erfolgt als String der wiederum aus Integer-Werten und Teil-Strings zusammengesetzt ist.

Das Programm besitzt keine Datenschnittstellen außerhalb seiner Laufzeitumgebung und erzeugt oder liest auch keine persistenten Daten.

Wichtige limitierende Größen sind als #define verfügbar und können so leicht gefunden und ggf. angepasst werden.

Entwurf

3.3 Systemaufbau

Das Gesamtprogramm wird gemäß dem Pflichtenheft aufgebaut. Die innere Struktur ist unter 3.4 weitergehend beschrieben.

3.3.1 Aufrufdiagramm der Grundversion

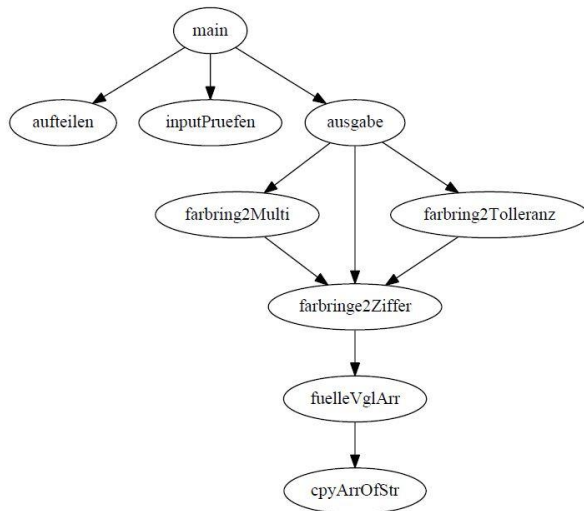


Abbildung 1: Aufrufdiagramm der Grundversion

3.3.2 Aufrufdiagramm der Vollversion

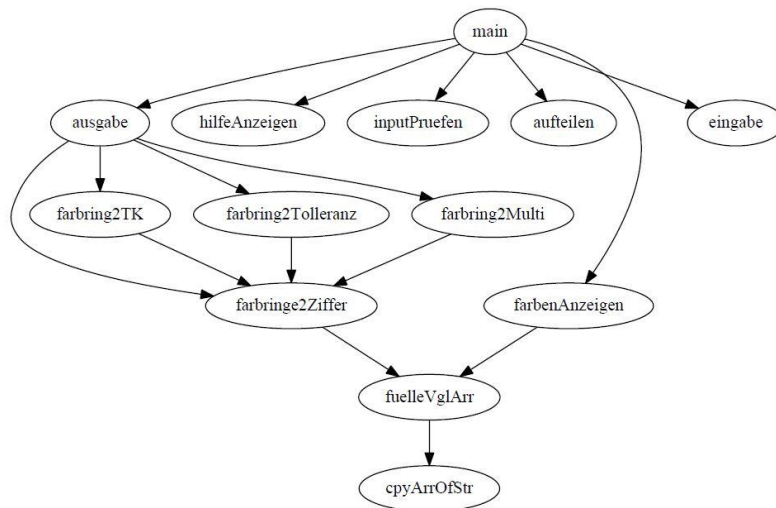


Abbildung 2: Aufrufdiagramm der Vollversion

3.4 Definition der Module

Die Module des Programms werden durch die einzelnen Funktionen definiert. Das Produkt ist in einer Source-Datei erstellt. Zum kompilieren sind nur Bibliotheken notwendig die zum Standardumfang der gängigen C-Compiler gehören. Funktionalitäten von Drittanbietern werden nicht verwendet.

Für das Grundprogramm werden die Prototypen gemäß C73A.3 verwendet (vgl. Abb. 1)

Entwurf

Für die Vollversion wird eine abgeänderte Struktur verwendet. Diese unterscheidet sich hauptsächlich in Ergänzten Funktionen (dementsprechend Funktionsaufrufen). (vgl. Abb. 2)

3.4.1 Eingabe der Widerstandsfarbringe

Die möglichen Eingaben beziehen sich auf folgende Tabelle:

Farbe	1.Band	2.Band	3.Band	Multiplikator	Toleranz	Temperatur-Koeffizient (10 ⁻⁶ /K) (PPM)
schwarz, sw, black, bk	0	0	0	x1		± 250
braun, br, brown, bn	1	1	1	x10	± 1%	± 100
rot, rt, red, rd	2	2	2	x100	± 2%	± 50
orange, or, og	3	3	3	x1k		± 15
gelb, ge, yellow, ye	4	4	4	x10k		± 25
gruen, gn, green, grün, grÜn, grun	5	5	5	x100k	± 0,5%	± 20
blau, bl, blue, bu	6	6	6	x1M	± 0,25%	± 10
violett, vi, violet, vt, lila, vio	7	7	7	x10M	± 0,10%	± 5
grau, gr, grey, gy	8	8	8	x100M	± 0,05%	± 1
weiss, ws, white, wh, weis	9	9	9	x1G		
gold, au, go, gd				x0.1	± 5%	
silber, si, silver, sr, ag				x0.01	± 10%	

Tabelle 1: Eingabe der Widerstandsfarbringe

Das einlesen der Eingabe beinhaltet die Wandlung nach Kleinbuchstaben. Damit ist Groß- und Kleinschreibung egal. Die Ausnahme bildet das ‚Ü‘. Zur Wandlung nach Kleinbuchstaben wird eine Funktion verwendet die keine Umlaute unterstützt.

3.4.2 Verarbeitung der Eingabe

Zuerst wird die Eingabe in ihre vier Farbringe (Programm-Grundversion) aufgeteilt. Danach werden die einzelnen Farbringe in ihre korrespondierenden Zahlen mit Hilfe eines 2-dimensionalen Arrays-of-String umgewandelt.

Die Form sieht Entwurfsweise folgendermaßen aus:

0	schwarz	sw	black	bk
1	braun	br	brown	bn
2	rot	rt	red	rd

Tabelle 2: Verarbeitung der Eingabe

Dabei stellt eine Länge des Arrays größer als die benötigte Zellenmenge eine einfache Erweiterbarkeit für weitere mögliche Farbbezeichnungen dar.

Mit Hilfe dieser Zahlen werden anschließend die passenden Multiplikatoren und Toleranzen gewählt.

Implementation + Test

3.4.3 Ausgabe des Widerstandswertes

Bei der Ausgabe werden 1000 Ohm zu 1 kOhm, 1000 kOhm zu 1 MOhm und 1000 MOhm zu 1 GOhm gewandelt.

Die Ausgabe des Widerstandswertes erfolgt dann im Muster: „4,7 kOhm +/-5%“.

Die Ausgabefunktion erreicht besonders in der Vollversion eine hohe Schachtelungstiefe. Zur Dokumentation ist ein Struktogramm (Anhang: PseudoStruktogramm Ausgabe.jpg) angelegt. Dieses gibt (über weite Teile im originalen C-Syntax) einen Überblick zur gesamten Funktion.

4 Implementation + Test

4.1 Implementation

Die Implementation erfolgt anhand dieser Dokumentation in ANSI-C. Die Lauffähigkeit ist auf UNIX (Ubuntu) und Win64- Systemen getestet. Für die jeweiligen Architekturen ist das Programm in der jeweiligen Umgebung zu kompilieren. Die Präcompiler-Direktiven (zur Bestimmung des Betriebssystems) beziehen sich auf Umgebungsvariablen die durch gcc definiert sind.

Als Entwicklungsumgebung werden verschiedene Software-Pakete benutzt. Diese sind nicht Bestandteil dieses Produktes jedoch frei verfügbar. Auf Windows-Betriebssystemen wird VisualStudio 2017 -CE und VisualStudio Code verwendet. Unter Linux (Ubuntu/ debian) wird gedit verwendet.

Als Compiler dient jeweils gcc. Unter Windows wird als Paket MinGW zur Bereitstellung verwendet. (Siehe Programmverzeichnis)

4.2 Weitere Software

Zur (grafischen) Dokumentation werden weitere Softwarepakete verwendet. Auch hier wurde auf frei verfügbare Software (bzw. frei für Bildungszwecke) Wert gelegt. (Siehe Programmverzeichnis)

4.3 Hauptprogramme:

- Grundversion
- Vollversion

4.4 Testprogramme:

- Aufteilen
- Farbring2Tol
- Farbringe2Multi
- Farbringe2Ziffer

Einführung

4.5 Nachweis der Funktionalität

Die Funktionalität des Gesamtproduktes sowie der einzelnen Bestandteile sind sorgfältig und ausführlich getestet. Diese Tests kontrollieren die Richtigkeit der Ausgaben sowie das Verhalten bei Fehleingaben.

Die Testprogramme (als Produktbestandteil) prüfen die gängigsten Anwendungsfälle ab. Weitere Eingaben können innerhalb der Testprogramme durch den Benutzer getätigt werden.

4.5.1 Disclaimer

Die Software sowie ihre Bestandteile sind ausführlich getestet. Jedoch kann es zu unerwarteten Fehler kommen die einen Programmabsturz erzeugen. Es gilt keine Verbindlichkeit auf die Richtigkeit der Ergebnisse und auch die Interpretation der Ausgaben ist dem Benutzer vorbehalten. Im Allgemeinen gilt immer die GMV-Richtlinie. (Gesunder-Menschen-Verstand-Richtlinie).

5 Einführung

Die Einführung ist mit der Aushändigung des Datenträgers dieser Dokumentation abgeschlossen.

5.1 Lieferumfang

1x Bedienungsanleitung

1x Dokumentation

1x Der Widerstandsrechner als C-Programm (Grundversion)

1x Der Widerstandsrechner als C-Programm (Vollversion)

1x Die Testprogramme (Aufteilen, Farbring2Tol, Farbringe2Multi, Farbringe2Ziffer)

Diverse Dokumentationen

Source-Dateien der Dokumentationsbestandteile

6 Persönliche Erklärung

Hiermit bestätigen wir, dass die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt wurden. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach deren Werken (dazu zählen auch Internetquellen) entnommen sind, wurden unter Angaben der Quellen kenntlich gemacht.

Zusatz für Software: Hiermit bestätigen wir, dass der Sourcecode und dessen Algorithmik von uns entwickelt ist. Verwendete Fremdpakete sind entsprechend gekennzeichnet. Falls bekannte Algorithmen (z.B. Suchverfahren) verwendet sind, werden diese mit Quelle gekennzeichnet.

Steinhagen, <Datum>

Bielefeld, <Datum>

7 Abbildungsverzeichnis

Abbildung 1: Aufrufdiagramm der Grundversion	4
Abbildung 2: Aufrufdiagramm der Vollversion	4

8 Tabellenverzeichnis

Tabelle 1: Eingabe der Widerstandsfarbringe.....	5
Tabelle 2: Verarbeitung der Eingabe.....	5

9 Quellenverzeichnis

DIN und IEC-Normen (nach Wikipedia zitiert)

Westermann Elektronik Tabellenbuch für Betriebs- und Automatisierungstechnik (1. Auflage 2011)
Seite 46

<http://www.gyrator.de/material/> für die Anhänge C73A.pdf und O31F.pdf

Für Anregungen bei der Implementation: <https://stackoverflow.com>

10 Softwareverzeichnis

10.1 IDEs

- (Gedit)
- Microsoft Visual Studio Code (Version 1.21.1)
- Microsoft Visual Studio 2017 CE-Edition (Version 15.6.4)

10.2 Dokumentation

- Microsoft Word 2016 (365)
- DIA (Version 0.97.2) [Anwendung zum Zeichnen von strukturierten Diagrammen]
- egypt (Version 1.10) [Anwendung zur Erstellung von Ablaufdiagrammen]
(<http://www.gson.org/egypt/>)
 - o zusätzlich die benötigten Abhängigkeiten Perl und Graphviz
- Okular (als Teil einer Ubuntu Dist.) [Anwendung für .ps zu .pdf]

Softwareverzeichnis

- Struktogrammeditor (Version 1.7) [Anwendung für [Struktogramme]
(<http://www.whiledo.de/index.php?p=struktogrammeditor>)

10.3 Datenverwaltung

- Microsoft OneDrive
- git
- gitHub
- SourceTree