



► **Prüfungsbeispiel:**
Zusatzqualifikation
Programmierung
zu Kapitel 5

zu:

AUSBILDUNG GESTALTEN:

Industrielle Elektroberufe

Mechatroniker und Mechatronikerin.

Umsetzungshilfen und Praxistipps.

Hrsg.: BIBB. Bonn 2018

Prüfungsbeispiel: Zusatzqualifikation Programmierung

Die Zusatzqualifikation Programmierung ist eine der Zusatzqualifikationen für die Berufe

- Elektroniker/in für Automatisierungstechnik
- Elektroniker/in für Betriebstechnik
- Elektroniker/in für Geräte und Systeme
- Elektroniker/in für Informations- und Systemtechnik
- Elektroniker/in für Gebäude und Infrastruktursysteme
- Mechatroniker/in

Das vorliegende Prüfungsbeispiel wurde von einem Auszubildenden im Beruf Elektroniker/in für Informations- und Systemtechnik bearbeitet, der in einem städtischen Verkehrsbetrieb ausgebildet wird.

Das Prüfungsbeispiel besteht aus folgenden Bestandteilen:

1. **Deckblatt für den Report** (sollte von der prüfenden IHK als Vorlage zur Verfügung gestellt werden)
2. **Report mit Anlagen** (wurde vom Prüfling im Rahmen der Bearbeitung der praxisbezogenen Aufgabe erstellt und dem Prüfungsausschuss vor der Prüfung vorgelegt)
3. **Leitfaden für das Fachgespräch** (vom Prüfungsausschuss vor der Prüfung vorbereitet und während des Fachgesprächs verwendet)
4. **Gesamtbewertungsbogen** (als Muster von der IHK bereitgestellt und vom Prüfungsausschuss vorbereitet; Hinweis: Die Gewichtungsfaktoren sind hier noch nicht eingefügt. Sie sollten im Anwendungsfall ebenfalls vor dem Fachgespräch vom Prüfungsausschuss festgelegt werden.)

IHK-LOGO**Deckblatt für den Report**

Bitte beachten Sie die formalen und inhaltlichen Hinweise zur Erstellung der Reporte.

Prüfungsteilnehmer
 Frau Herr
Ausbildungsbetrieb

 Name, Vorname

 Straße, Hausnummer

 PLZ

 Ort

 Geburtsdatum

 Geburtsort

 Telefon privat

 E-Mail

 Name und Anschrift des Ausbildungsbetriebes

 Ausbilder
Ausbildungsberuf:
Abschlussprüfung Teil 2 Sommer Winter des Jahres _____

Gewählte Zusatzqualifikation: (bitte kreuzen Sie eine an)

- | | |
|--|---|
| <input type="checkbox"/> 1. Systemintegration | <input type="checkbox"/> 5. Programmierung |
| <input type="checkbox"/> 2. Prozessintegration | <input type="checkbox"/> 6. Digitale Vernetzung |
| <input type="checkbox"/> 3. Additive Fertigungsverfahren | <input type="checkbox"/> 7. IT-Sicherheit |
| <input type="checkbox"/> 4. IT-gestützte Anlagenänderung | |

Thema der praxisbezogenen Aufgabe / des Reports:

IBIS-Slave-Simulator

Ich versichere, die praxisbezogene Aufgabe selbstständig durchgeführt und den Report persönlich erstellt zu haben. Des Weiteren versichere ich, weder Inhalte noch teilweise oder vollständige Passagen aus anderen Reporten übernommen zu haben, die bei der prüfenden oder einer anderen IHK eingereicht wurden.

Ich bestätige, dass der/die o.g. Prüfungsteilnehmer/ Prüfungsteilnehmerin die praxisbezogene Aufgabe selbstständig durchgeführt und den Report persönlich erstellt hat.

 Ort, Datum, Unterschrift des Prüfungsteilnehmers

 Ort, Datum, Unterschrift/Stempel des Ausbildungsbetriebes

Inhaltsverzeichnis

1. Aufgabenstellung / Zielsetzung.....	2
2. Information und Planung	2
3. Vorgehensweise.....	3
4. Ergebnis der praxisbezogenen Aufgabe.....	4
5. Bewertung des Prozesses und des Ergebnisses.....	4
Anhang.....	5
Struktogramme zu den Funktionen	

Name, Vorname

Prüflingsnummer: xxyy

Datum

Analysephase

1. Aufgabenstellung / Zielsetzung

Während meiner Zeit in der zentralen Elektronik-Werkstatt der Verkehrsbetriebe in der Seestraße bestand aufgrund eines aktuellen Auftrags Interesse an einem Gerät, welches die gesendeten Datenpakete der in Bussen und Bahnen installierten Überwachungskameras auf dem IBIS – Bus (<https://www.trapezgroup.de/de/linienverkehr/details/bordrechner/>) anzeigt und auf Slave-Aufrufe reagiert. Damit sollte es möglich sein, das Verhalten der Teilnehmer und des Masters besser nachzuvollziehen. Daraus ergab sich ein Projekt mit folgenden Anforderungen:

- die Erstellung eines Gerätes zur Simulation von Slaves auf dem IBIS – Bus
- Anzeige der auf dem Bus gesendeten Datenpakete (vorrangig vom Master) auf dem PC über ein Terminal Programm (z.B. Putty)
- alle Datenpakete sollen erfasst und am PC wiedergegeben werden können
- auf Statusabfragen bestimmter Teilnehmer („a2“ – „a?“) soll mit einem festgelegten Status geantwortet werden
- Änderung des Status eines Teilnehmers vom PC aus
- Filterung der anzuzeigenden Datenpakete
- einstellbare Zeitverzögerung zwischen Aufruf des Masters und der Antwort

2. Information und Planung

Nach Erhalt des Arbeitsauftrages kristallisierten sich einige Fragen besonders heraus.

- Was ist IBIS?
- Wie kommuniziert der Bus?
- Wie sieht ein Telegramm aus?
- Welche Hardware kann ich nutzen?
- Wie und womit wird programmiert?

Anhand dieser Fragen habe ich mich dann in das Thema eingearbeitet und konnte folgende Feststellungen treffen.

Der IBIS – Bus (IBIS = Integriertes Boardinformationssystem) ist ein Bussystem, welches vorrangig in öffentlichen Verkehrsmitteln verwendet wird, um Daten wie: Uhrzeit, Datum, Linie, Fahrtrichtung und den Status verschiedener Anzeigen wiederzugeben. Der Bus besteht aus einer Datenleitung für den Master, zum Senden von Daten und Abfragen der Teilnehmer und zugehöriger Masseleitung und einer Datenleitung für die Slaves, zum Antworten auf Abfragen, mit zugehöriger Masseleitung. Die Abfrage der Teilnehmer erfolgte im Polling Verfahren. Die Einstellungen sind: 1200Baud, 7 Datenbit, Gerade Parität (Even Parity) und 2 Stopbits. Als Besonderheit werden die Datenpakete auf dem

Bus mit einer XOR - Checksumme übertragen. Außerdem arbeitet der Bus mit 24Volt. Daher benötigte ich eine Anpassungsschaltung auf 5Volt für den Mikrocontroller. Diese Schaltung existierte bereits in ähnlicher Weise und wurde mir in Form eines Schaltplans von meinem Auftraggeber übergeben. Durch den Wunsch des Auftraggebers, in einer späteren Geräteversion die gesendeten Daten auf der Slave-Leitung abzuhören, musste ich den Schaltplan anpassen. Aufgrund der Anforderung an den Mikrocontroller von 2 USART Schnittstellen, der Verfügbarkeit und der Baugröße fiel die Wahl auf ein Atmega128 – USB Shield (siehe Entscheidungsmatrix, Abb. 1).

	Arduino Uno	Atmega128 USB-Shield	Atmega328PB
min. 2 USART Schnittstellen	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
bereits vor Ort	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
mit Zusatzschaltung	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
kompakte Bauform	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Abbildung 1: Entscheidungsmatrix zur Controller Wahl

Bei diesem ist außerdem die USART1 Schnittstelle bereits an eine Mini-USB Buchse angeschlossen und kann direkt zur Kommunikation mit dem PC genutzt werden. Die verbleibende USART Schnittstelle konnte ich über die Anpassungsschaltung an den IBIS – Bus anschließen.

Zum Programmieren habe ich einen „AVRISP mkII“ verwendet und die Umgebung „Atmel Studio 7.0“, aufgrund der umfangreichen Möglichkeiten zum Simulieren und zur Fehlersuche. Als Programmiersprache habe ich, aufgrund meiner eigenen Kenntnisse und durch die Vorgabe des Auftraggebers, C genutzt.

3. Vorgehensweise

Da der IBIS – Bus mit 24 Volt Spannung arbeitet und der ausgewählte Mikrocontroller mit 5 Volt, habe ich nach der Rücksprache mit dem Auftraggeber zunächst eine Testschaltung für die galvanisch getrennte Sende- und Empfangsschaltung auf einem Testboard aufgebaut. Mithilfe einer einfachen Übertragung von einem Zeichen, konnte ich die Schaltung probeweise auf Ihre Funktion testen. Nach erfolgreichem Test der Hardware, habe ich einen Ablaufplan für das eigentliche Programm erstellt. Dabei konnte ich das Programm in einzelne Teilschritte / Funktionen zerlegen, für welche ich dann Struktogramme erstellt habe. Nachdem der Simulator Datenpakete erkannt und auf der PC– Schnittstelle wiedergegeben hat, habe ich eine geeignete Platine entwickelt. Dafür waren besonders die Größe des Gehäuses und die Bauform entscheidend, die eine bestimmte Anordnung der Bauteile auf der Platine vorgab. Nachdem ich das Gehäuse ausgemessen habe, habe ich ein geeignetes Layout erstellt und mit einer Fräsmaschine

Umsetzungsphase

fräsen lassen. Die Besonderheit hierbei, war die exakte Ausrichtung der Platine, da sie beidseitig ein Layout enthält. Anschließend habe ich die Aussparungen für die Schnittstellen in die Gehäuseabdeckungen gebohrt und gefeilt.

Nach Abschluss des Gehäuses, habe ich die Nutzersteuerung implementiert, da diese die geringste Priorität einnahm.

4. Ergebnis der praxisbezogenen Aufgabe

Name, Vorname

Prüflingsnummer: xxyy

Datum

Der Simulator ermöglicht nun eine Anzeige der auf dem IBIS – Bus gesendeten Daten am PC. Das Gerät befindet sich in einem kompakten, robusten Metallgehäuse. Der Anwender kann einfach über den PC den Status von Teilnehmern verändern oder anzeigen lassen und die Datenpakete filtern. Es ist auch möglich die Zeit zwischen Statusabfrage und -antwort einzustellen. Damit ist eine detailliertere Analyse des Busses und seiner Teilnehmer als bisher möglich.

5. Bewertung des Prozesses und des Ergebnisses

Durch die umfangreiche Planungsphase und die Auskünfte von meinem Auftraggeber gestaltete sich die Abarbeitung des Projektes sehr gut. Das Erstellen des doppelseitigen Layouts für die Platine war für mich eine ungewohnte Arbeit, weil ich bis dahin vermehrt einseitige Layouts erstellt und noch nie mit einer Fräsmaschine gearbeitet hatte. Durch die Betreuung der Kollegen vor Ort und deren Hilfe, stellte aber auch dies absolut kein Problem dar. Ebenso half es bei der Programmierung sehr, dass ich im Voraus bereits einen Ablaufplan und Struktogramme erstellt habe. Eine besondere Schwierigkeit war zunächst die Bildung der Prüfsumme. Nach einiger Recherche, gelang mir diese Aufgabe jedoch sehr gut. Ebenso war die Auswertung der vom Nutzer übergebenen Befehle eine Herausforderung, weil sie viel Zeit beanspruchte. Letztlich gelingt die Auswertung der Befehle zwar, der Programmierstil gefällt mir jedoch nicht. Ebenso wäre die Speicherverwaltung verbesserungswürdig, indem an einigen Stellen Pointer statt Variablen verwendet werden.

Insgesamt bin ich sehr zufrieden mit dem Ergebnis. Ich konnte nicht nur meine Fähigkeiten in der Planung und Durchführung von Projekten, sondern auch in der Programmierung und im Handwerk verbessern. Besonders der Nutzen eines Programmablaufplans und eines Struktogrammes hat sich gezeigt.

Zu meiner Freude wurde das Gerät bereits nach kurzer Zeit in Betrieb genommen und eingesetzt. Momentan befindet es sich im weiteren Ausbau.

Name, Vorname

Prüflingsnummer: xxyy

Datum

Anhang

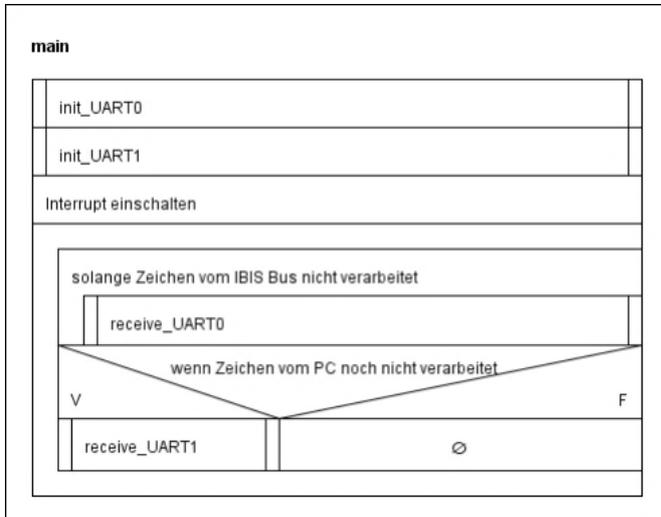


Abbildung 2: Struktogramm - main Funktion



Abbildung 3: Struktogramm - init-UART0 Funktion

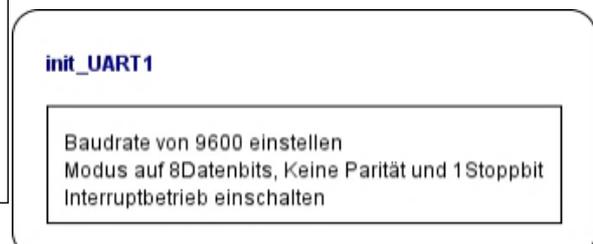


Abbildung 4: Struktogramm - init-UART1 Funktion

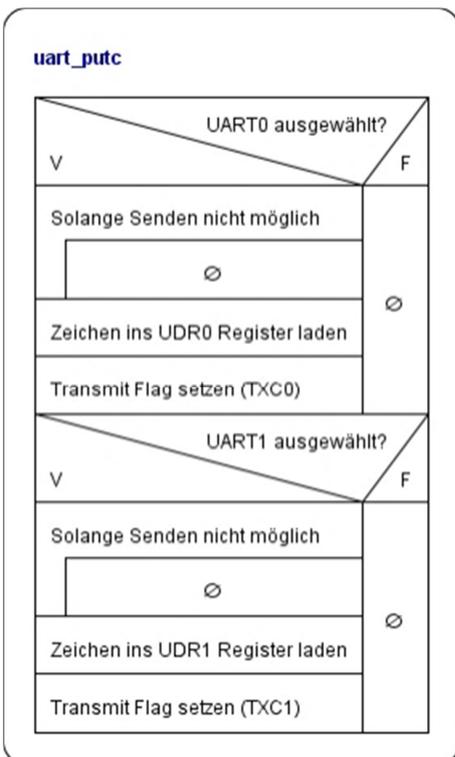


Abbildung 5: Struktogramm - uart_putc Funktion

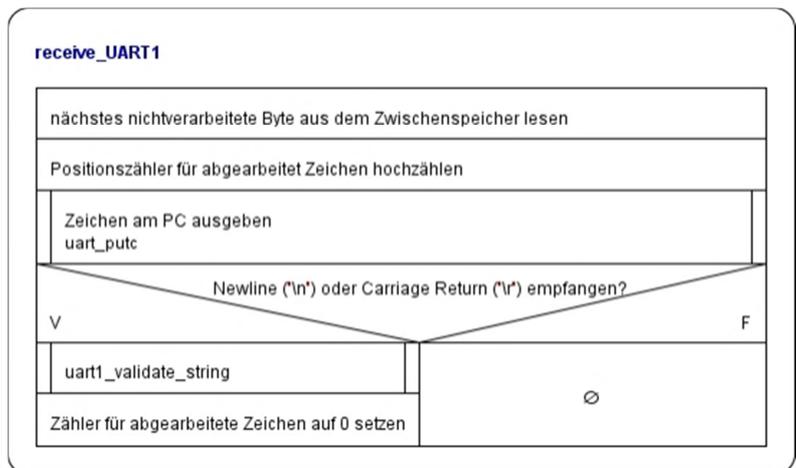


Abbildung 6: Struktogramm - receive_UART1 Funktion

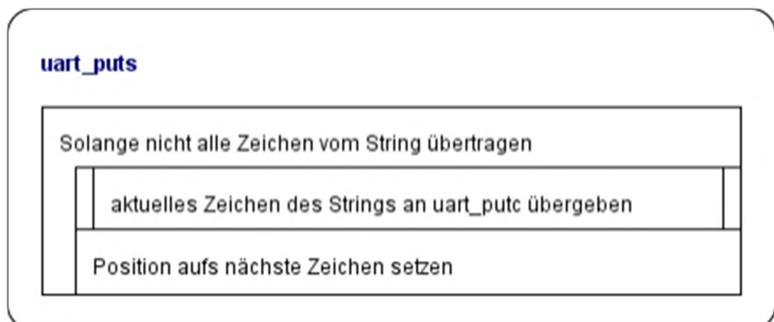


Abbildung 7: Struktogramm - uart_puts Funktion

Name, Vorname

Prüfungsnummer: xxyy

Datum

Anhang

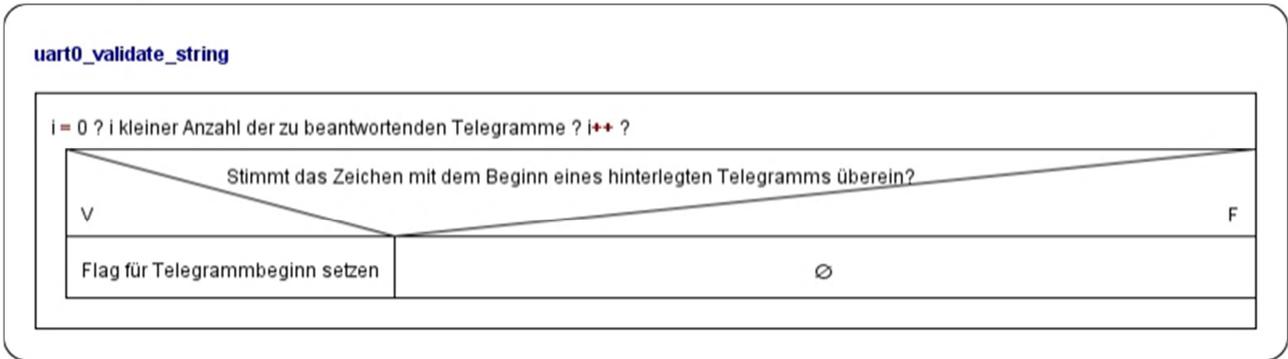


Abbildung 8: Struktogramm - uart0_validate_string

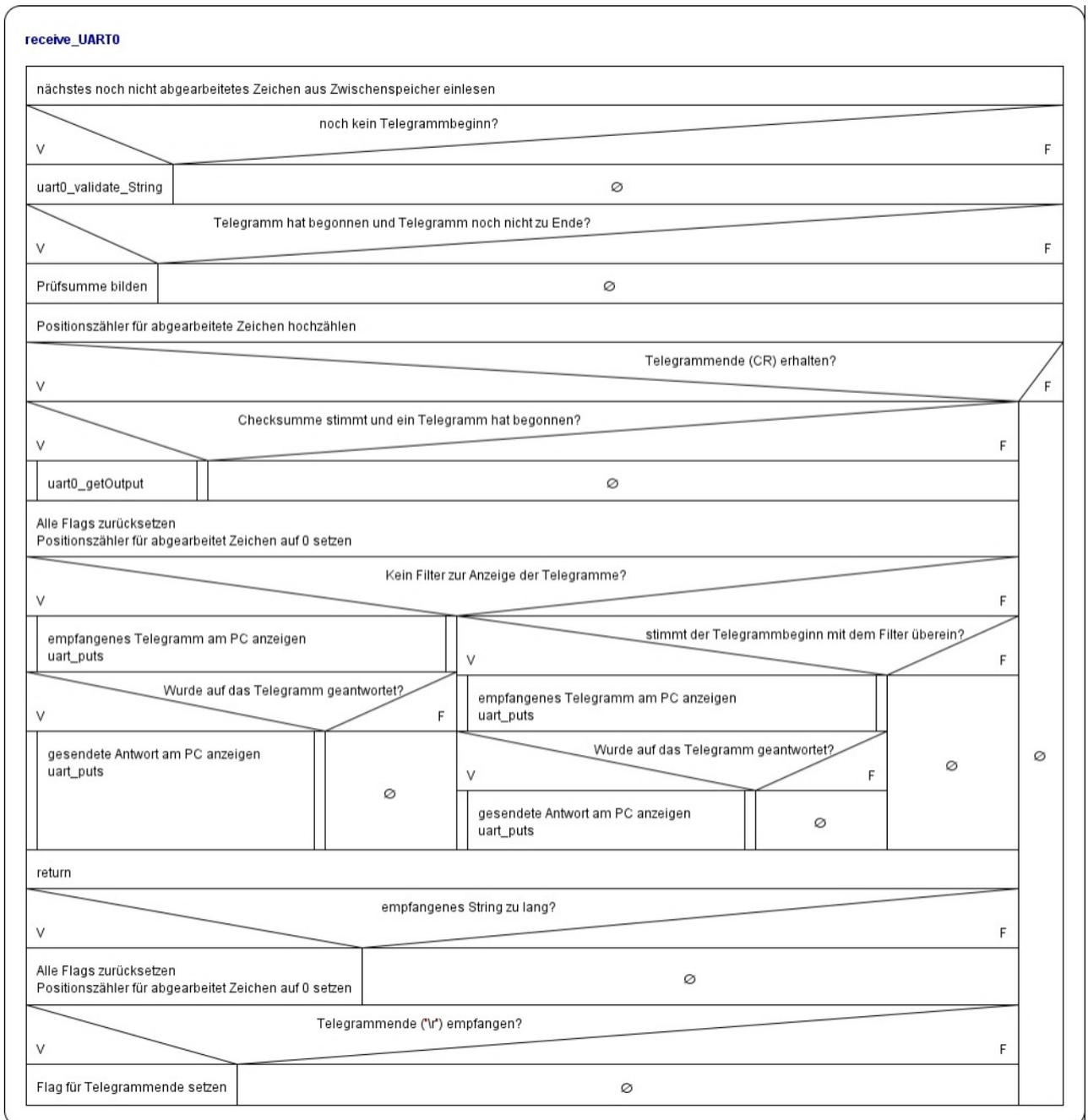


Abbildung 9: Struktogramm – receive_UART0 Funktion

Name, Vorname

Prüflingsnummer: xxyy

Datum

IBIS – Slave Simulator

Anhang

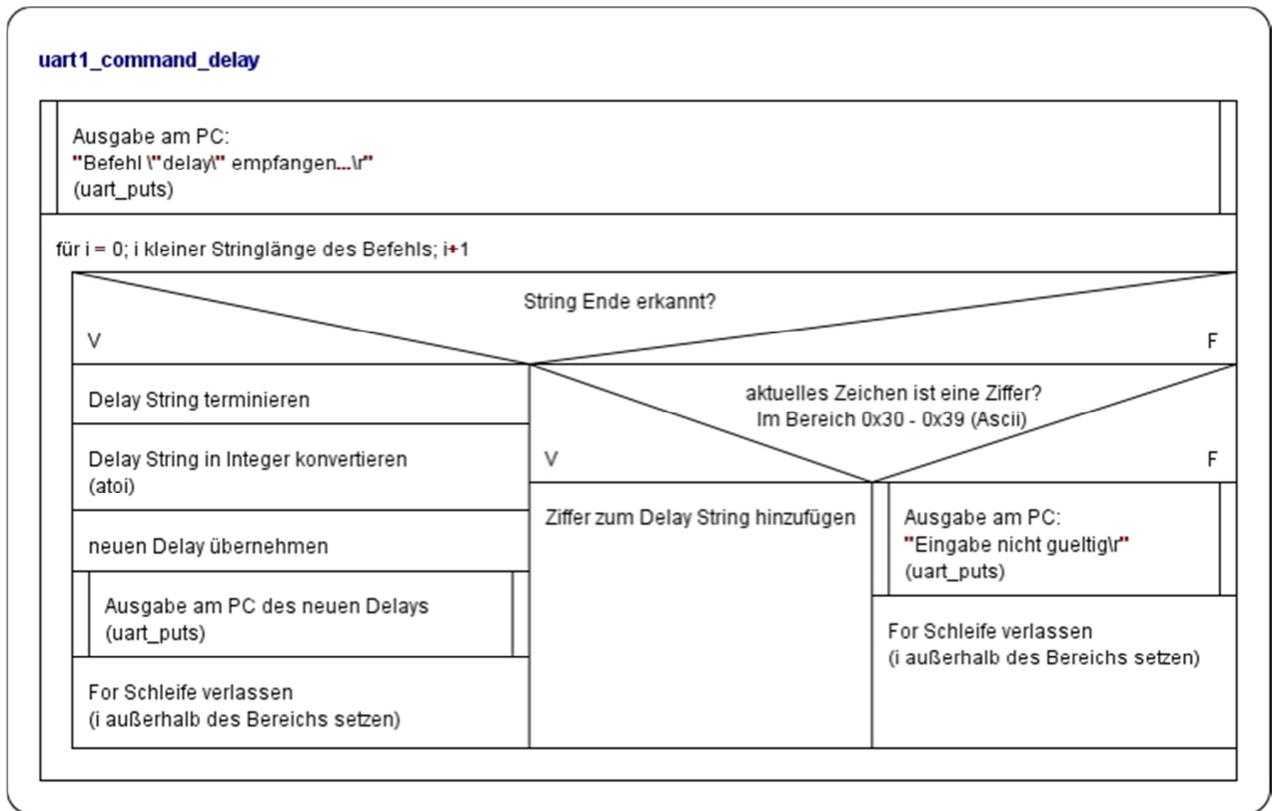


Abbildung 10: Struktogramm - uart1_command_delay

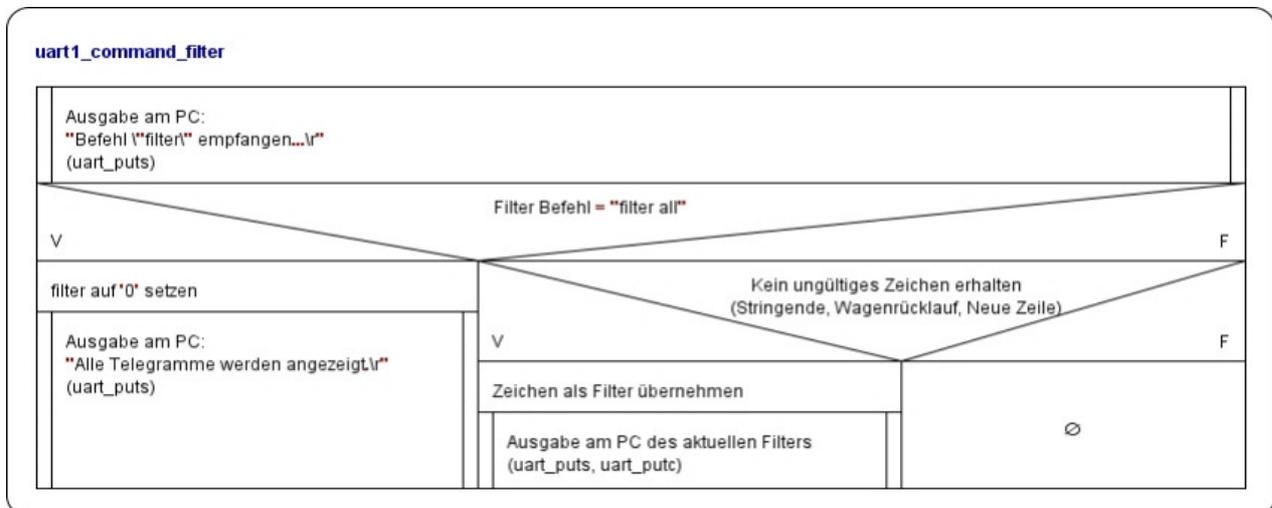


Abbildung 11: Struktogramm - uart1_command_filter

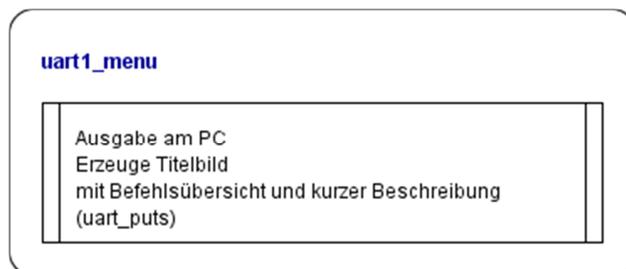


Abbildung 12: Struktogramm - uart1_menu

Name, Vorname

Prüfungsnummer: xxyy

Datum

IHK Abschlussprüfung Teil 2 – Winter 2020/2021	Leitfaden für das Fachgespräch
Zusatzqualifikation Programmierung	Ausbildungsberuf: Elektroniker/in für Informations- und Systemtechnik

Name des Prüflings:

Datum:

Titel der praktischen Aufgabe:

Analysieren von technischen Aufträgen und Entwickeln von Lösungen (Zeitrichtwert: 8 min)

- Erläutern Sie den (komplexen) Arbeitsauftrag!
- Welche Schnittstellen haben Sie verwendet? Erläutern Sie diese?
- Mit welchen Problemen waren Sie aus Ihrer Sicht zu Beginn des Projekts konfrontiert?
- Warum haben Sie als Programmiersprache C gewählt?
- Sind aufgrund der Wahl der Programmiersprachen auf Probleme gestoßen? Wenn ja, welche?

Anpassen von Softwaremodulen (Zeitrichtwert: 4 min)

- Wie haben Sie Ihre Softwaremodule dokumentiert?
- Wie haben Sie Ihre Softwaremodule in das vorhandene System integriert?

Testen von Softwaremodulen (Zeitrichtwert: 8 min)

- Wie haben Sie die einzelnen Softwarekomponenten getestet?
- Wie haben Sie den Test des gesamten Systems durchgeführt?
(Stichworte: Anwendungstest, Belastungstest, Testfälle, Testphase)
- In der Bewertung führen Sie Probleme bzw. und verbesserungswürdige Lösungen auf (Prüfsumme, Programmierstil, Speicherverwaltung). Erläutern Sie exemplarisch einen dieser Punkte.

