



SOFTWARE- ENTWICKLUNGSPRAKTIKUM (SEP)

ARID - AUGMENTED REALITY IN DISGUISE

Software-Entwicklungspraktikum (SEP)
Sommersemester 2024

Testspezifikation

Auftraggeber:

Technische Universität Braunschweig
Institut für Anwendungssicherheit (IAS)
Prof. Dr. Martin Johns
Mühlenpfordstraße 23
38106 Braunschweig

Betreuerin: Alexandra Dirksen

Auftragnehmer:

Name	E-Mail-Adresse
Amir Fakhim Hashemi	a.fakhim-hashemi@tu-braunschweig.de
Ibrahim Abdullah	i.abdullah@tu-braunschweig.de
Jadon-Kim Fischer	jadon-kim.fischer@tu-braunschweig.de
Mohamed Ali Mrabti	m.mrabti@tu-braunschweig.de
Tim Küttemeyer	t.kuetemeyer@tu-braunschweig.de
Vyvy Nguyen	Vyvy.nguyen@tu-braunschweig.de

Braunschweig, 3. Juli 2024

Bearbeiterübersicht

Kapitel	Autoren	Kommentare
1	Ibrahim	—
2	Vyvy	—
2.1	Vyvy	—
2.2	Vyvy	—
2.3	Vyvy	—
2.4	Vyvy	—
2.5	Vyvy	—
3	Tim	—
3.1	Tim	—
3.2	Tim	—
3.3	Tim	—
4	Amir, Vyvy	—
4.1	Amir, Vyvy	—
4.2	Amir	—
4.3	Amir	—
5	Mohamed Ali	—
5.1	Mohamed Ali	—
5.2	Mohamed Ali	—
5.3	Mohamed Ali	—
6	Sämtliche	Fortlaufend ergänzt

Inhaltsverzeichnis

1	Einleitung	6
2	Testplan	7
2.1	Zu testende Komponenten	7
2.2	Zu testende Funktionen/Merkmale	8
2.3	Nicht zu testende Funktionen	8
2.4	Vorgehen	9
2.5	Testumgebung	11
3	Abnahmetest	12
3.1	Zu testende Anforderungen	12
3.2	Testverfahren	13
3.2.1	Testskripte	13
3.3	Testfälle	13
3.3.1	Testfall $\langle T100 \rangle$ - Power On/Off	14
3.3.2	Testfall $\langle T200 \rangle$ - Bluetooth-Konnektivität	15
3.3.3	Testfall $\langle T300 \rangle$ - Touchpad-Reaktivitäts	16
3.3.4	Testfall $\langle T400 \rangle$ - QR-Code-Erkennung	17
3.3.5	Testfall $\langle T500 \rangle$ - Entschlüsselung von QR-Codes	18
3.3.6	Testfall $\langle T600 \rangle$ - Erkennung verschlüsselter QR-Codes	19
3.3.7	Testfall $\langle T700 \rangle$ - Anzeigen der Nachricht	20
3.3.8	Testfall $\langle T800 \rangle$ - Funktionalität der Kamera	21
4	Integrationstest	22
4.1	Zu testende Komponenten	22
4.2	Testverfahren	22
4.2.1	Testskripte	23
4.3	Testfälle	23
4.3.1	Testfall $\langle T200 \rangle$ - Komponente C00+C10+C40+C50+C60+C70	24
4.3.2	Testfall $\langle T210 \rangle$ - Komponente C20 + C30	25
4.3.3	Testfall $\langle T220 \rangle$ - Komponente C60 + C70	26

5	Unit-Tests	27
5.1	Zu testende Komponenten	27
5.2	Testverfahren	28
5.2.1	Testskripte	28
5.3	Testfälle	28
5.3.1	Testfall $\langle T300 \rangle$ - Ein -und Ausschaltfunktion des Monocle	29
5.3.2	Testfall $\langle T301 \rangle$ - Kommunikation zwischen dem Endgerät und dem Monocle	30
5.3.3	Testfall $\langle T302 \rangle$ - Touchpad	31
5.3.4	Testfall $\langle T303 \rangle$ - QR-Code Lesen	32
5.3.5	Testfall $\langle T304 \rangle$ - Entschlüsselung von QR-Codes	33
5.3.6	Testfall $\langle T305 \rangle$ - Erkennung verschlüsselter QR-Codes	34
5.3.7	Testfall $\langle T306 \rangle$ - Anzeigen der Nachrichten	35
5.3.8	Testfall $\langle T307 \rangle$ - Kamera	36
6	Glossar	37

Abbildungsverzeichnis

1 Einleitung

Diese Testspezifikation dokumentiert die geplanten und durchgeführten Tests fürs System. Ziel ist es, die Funktionalität und die Leistung des Systems zu überprüfen und sicherzustellen, dass alle Anforderungen und Spezifikationen erfüllt werden. Die Tests werden basierend auf den definierten Testfällen durchgeführt, um die Qualität des Produkts zu sichern.

Dabei werden die zu testenden Softwarekomponenten und -funktionen beschrieben, die Testumgebung definiert sowie detaillierte Testfälle und das Vorgehen für verschiedene Testarten, einschließlich Integrationstests, Abnahmetests und Unit-Tests, festgelegt. Für die Integrationstests zielen darauf ab, die reibungslose Zusammenarbeit der einzelnen Komponenten zu überprüfen. Der Abnahmetest dient als Beleg für die Erfüllung der zwischen Kunde und Auftragnehmer vereinbarten Leistungen und stellt die Produktvollständigkeit bei der Übergabe sicher. Die Unit-Tests testen die einzelnen Softwarekomponenten isoliert, um deren korrekte Funktion zu sichern.

Zum testende Software gehört das Testen der automatischen Ein- und Ausschaltfunktion des Monocle stellt sicher, dass das Gerät zuverlässig ein- und ausgeschaltet werden kann. Die BLE-Kommunikation zwischen dem Endgerät und dem Monocle wird überprüft, um eine stabile und fehlerfreie Datenübertragung zu sichern. Auch die Funktionalität der Touchpads wird getestet, dass sie auf Benutzereingaben korrekt reagieren. Das Lesen des QR-Codes wird überprüft, um sicherzustellen, dass der QR-Code fehlerfrei erkannt wird. Außerdem wird die Verschlüsselung von Nachrichten getestet, um die Sicherheit übertragenen Daten zu sichern. Das Anzeigen der Nachricht wird geprüft, um sicherzustellen, dass Nachrichten korrekt und lesbar auf dem Display des Monocle dargestellt werden. Schließlich wird die Kamera getestet, dass sie klare Bilder aufnimmt.

Das Testen ist dem IEEE 829-Standard angelehnt, um ein zuverlässiges und funktionierendes Produkt bereitzustellen. Es sollen durch die ganzen Test Probleme identifiziert werden und möglichst auch behoben. Trotz der umfangreichen Tests können Fehler und Irrtümer auftauchen.

2 Testplan

Im Kapitel des Testplans wird die Qualitätssicherung des Produktes gewährleistet. Es wird dabei auf den Umfang und die Vorgehensweise der Qualitätssicherung eingegangen. Analysiert werden dafür die Testgegenstände, deren Eigenschaften und Funktionen. Außerdem werden die Personen, die die Maßnahmen durchführen, genannt.

Folgende Abschnitte sind dem Abnahmetestspezifikation überarbeitet entnommen worden.

2.1 Zu testende Komponenten

Im folgenden Abschnitt werden die Komponenten aufgelistet, die im Rahmen der Tests berücksichtigt werden:

- **<C00> FPGA:** Die von der 5-MP-Kamera aufgenommenen Grafiken und Fotos werden durch den FPGA im Monocle-Gerät verbessert. Er verfügt über ein Gowin GW1N-LV9MG100C6/I5-Modell mit 608kb Flash-Speicher, 468kb Block-RAM und 7k LUTs. Zur Unterstützung beim FPGA-Start oder bei der Datenspeicherung ist ein zusätzlicher 1MB SPI Flash IC enthalten.
- **<C10> Host-Anwendung:** Es handelt sich hierbei um ein Mircopython Programm des Host, dessen Aufgabe die Kommunikation zwischen Host und Monocle und die Weiterverarbeitung der Informationen gewährleistet.
- **<C20> QR-Code-Erkennung:** Prozedur innerhalb desselben Skripts wie die Host-Anwendung, die jedoch unter Zuhilfenahme einer Software-Bibliothek von dritten umgesetzt wurde. Diese ist wichtig für die Optimierung der QR-Code Erkennung.
- **<C30> Entschlüsselung-Programm:** Ist ein Python Programm die auf dem Host Gerät läuft.
- **<C40> Monocle-Anwendung:** Programm des Monocles, dieser verknüpft die Funktionen des Monocle-Systems. Das Programm läuft auf dem Host wird jedoch über BLE an das Monocle versendet.

- **<C50> Bluetooth-MCU:** Die Haupt-CPU vom Monocle ist die Bluetooth-MCU. Der Nordic nRF52832-Speicher ist die verwendete MCU, die Bluetooth 5.2 unterstützt.
- **<C60> Kamera:** Die Omnivision OV5640 Frontkamera des Monocle ist direkt mit dem FPGA verbunden.
- **<C70> Touchpads:** Ein Azoteq IQS620A Touch Controller empfängt Eingaben von den Touchpads A und B oben auf dem Monocle-Gerät.
- **<C80> Display:** Ist ein 0.23" micro OLED, weißt 640x400 RGB pixels auf. (Siehe Pflichtenheft, Abschnitt 8.2)

2.2 Zu testende Funktionen/Merkmale

Nachstehend finden Sie eine Liste der Merkmale und Funktionen, die getestet werden müssen (sehen Sie auch Pflichtenheft Abschnitt 8):

- **<F10>** Testen der automatischen Ein- und Ausschaltfunktion des Monocle.
- **<F20>** Testen der BLE-Kommunikation zwischen Endgerät und Monocle.
- **<F30>** Testen der Touchpads.
- **<F40>** Lesen des QR-Codes.
- **<F50>** Testen der Verschlüsselung von Nachrichten.
- **<F60>** Anzeigen der Nachricht.
- **<F70>** Testen des Kamera.
- **<Q10>** Aktualisieren des Firmware.

2.3 Nicht zu testende Funktionen

In diesem Abschnitt werden wir die Funktionen auflisten, die im Test nicht abgedeckt werden, und die Gründe dafür nennen.

- **Geschwindigkeit des QR-Code-Scans :** Geschwindigkeit des QR-Code-Scans hängt von den Umgebungsbedingungen und die Qualität des QR-Codes ab. Deswegen werden diese nicht getestet.

- **Ladezeit des Codes** : Es gibt keine festen Kriterien für akzeptable Ladezeiten, da diese zu stark variieren.
- **Micro-/Python Libraries** : Diese Bibliotheken werden in der Regel als vertrauenswürdig und funktional betrachtet, deswegen werden diese nicht explizit getestet.
- **Kommunikationsgeschwindigkeit durch BLE** : Die Geschwindigkeit der Kommunikation durch Bluetooth Low Energy (BLE) wird nicht spezifisch getestet, da es an viele Umgebungsbedingungen liegt, wie zum Beispiel die Signalstärke, wie weit das Endgerät vom Monocle genutzt werden kann, und die Implementierung der BLE-Protokolle.
- **Ladezeit des Monocle** : Da ist es für die Benutzererfahrung als unwichtig erachtet wird, wird diese Funktion nicht getestet.

2.4 Vorgehen

Zum Testen der automatischen Ein- und Ausschaltfunktion des Monocle genügt es dieses aus seiner dafür vorgesehenen Hülle zu entfernen, beziehungsweise dorthin hineinzugeben. Ist das Gerät technisch funktionstüchtig und ausreichend geladen, sollte das Display des Geräts zu leuchten beginnen und der Schriftzug „Monocle“ sollte erkennbar sein. Da dieses Verhalten durch die interne Programmierung des Monocle-Prozessors bestimmt wird, handelt es sich um einen Black-Box-Test mit den möglichen Endzuständen Gerät aktiv und Gerät inaktiv, wobei bei dem Herausnehmen des Geräts der erste der beiden Zustände die gewünschte Funktionalität bestätigt und bei dem Hineingeben der letztere.

Für die BLE-Kommunikation zwischen Endgerät und Monocle muss das Monocle-Gerät per Bluetooth mit einem separaten PC oder Laptop verbunden werden. Eine zweite Verbindung findet durch den Programmcode statt, sobald das Monocle-Gerät gestartet und das Programm auf dem Laptop ausgeführt wurde. Die Auftragnehmer testen außerdem die Reichweite zwischen Gerät und Laptop, indem sie die räumliche Distanz zwischen beiden Geräten variieren und die Auswirkung auf die Verbindung und die davon abhängigen Funktionen (wie etwa die Bildübertragung) beobachten. Konkret verstehen die Teilnehmer die verschiedenen Distanzen zwischen Monocle und Host-Gerät als unterschiedliche Testphasen. Hierbei soll die Konfiguration bei der das Monocle-System unmittelbar auf dem Host-Gerät liegt als initiale Testphase betrachtet werden. Eine Testphase gilt als bestanden, sobald die Kommunikation erfolgreich und innerhalb einiger Sekunden abgewickelt wird.

Die am oberen Rand des Brillenglases befindlichen Touchpads testen die Teilnehmer bereits vor der Entwicklung der eigentlichen Anwendung. Hierfür nutzen sie eine Demo-Anwendung, dass bei Berührung des jeweiligen Touchpads eine entsprechende Nachricht in der Mitte des Displays anzeigt. Während der Entwicklung der eigenen Anwendung erkennen die Teilnehmer

schnell, ob die Funktionalität der Touchpads sichergestellt ist, da der dadurch ausgelöste Beginn des Einlesevorgangs mit einer dafür vorgesehenen Nachricht auf dem Display begleitet wird. De facto handelt es sich auch hier um einen Black-Box-Test, dessen Ausgang entweder durch die Erkennung der Eingabe erfolgreich ist oder unbekannt, da keine Eingabe über die Touch Controller registriert werden kann.

Wie bereits im Pflichtenheft beschrieben wird der Erfolg des Einlesevorgangs bereits von der Anwendung selber ermittelt. Je nach Ergebnis zeigt das Monocle-Gerät dann die in dem QR-Code verschlüsselte Nachricht oder eine Fehlermeldung an. Darüber hinaus sendet das Monocle-Gerät bei Aufnahme eines Fotos eben dieses Bild an den verbundenen Laptop zurück, sodass sich die Teilnehmer bereits im Voraus einen Eindruck der Bildqualität und der damit verbundenen Erfolgchance verschaffen können. Zum Testen der einzelnen Zwischenschritte, etwa die Speicherung der Bild-Aufnahme durch die Kamera im dafür vorgesehenen Zwischenspeicher, sollen White-Box-Tests zum Einsatz kommen.

Die Verschlüsselung der Nachrichten werden bereits im Vorfeld von den Teilnehmern auf ihren privaten Geräten getestet, da der dafür notwendige Programmcode den Teilnehmern als separate Dateien auf dem Git vorliegen. Da es keine nennenswerten Fälle für einen Blackbox-Test gibt, geben die Teilnehmer Botschaften unterschiedlicher Länge zur Verschlüsselung hinein und überprüfen im Anschluss daran die Entschlüsselung der erhaltenen Zeichenkette. Stimmt die entschlüsselte Botschaft mit der ursprünglichen überein, so war der Verschlüsselungsvorgang erfolgreich. Auf dem Monocle-Gerät folgen die Teilnehmer demselben Vorgehen, indem sie die entschlüsselte Botschaft auf dem Display betrachten. Da bei der Ver- und Entschlüsselung keinerlei Fallunterscheidungen in der Eingabe zu machen sind, testen die Teilnehmer diese dynamisch mit ihren eigenen Eingaben und erwarten, dass der algorithmische Ansatz auch statisch korrekt implementiert worden ist.

Ob die nach der Entschlüsselung angezeigte Nachricht auf korrekte Weise von dem Display des Monocle-Geräts dargestellt wird überprüfen die Auftragnehmer Hand in Hand mit der Verschlüsselung der Nachricht. Wird auf dem Gerät eine andere Nachricht angezeigt, als auf den Geräten der Teilnehmer, insbesondere nicht die Nachricht die in die Verschlüsselung eingegeben wurde, muss es sich dabei um einen Fehlerfall handeln. Dies lässt sich mit dem garantierten Aufruf der auf das Display bezogenen System-Funktion per White-Box-Test sicherstellen.

Zum Testen der Kamera greifen die Auftragnehmer auf die oben beschriebene Möglichkeit zurück, aufgenommene Bilder direkt an den mit dem Monocle-Gerät verbundenen Laptop zu senden. Ist eine mangelhafte Qualität der Aufnahme erkennbar (etwa durch Unschärfe oder Verfärbung), variieren die Teilnehmer die räumliche Position der Kamera, sowie die Struktur des fotografierten QR-Codes und dessen Beleuchtung. Sollten die Auftragnehmer unzureichende Qualität der Bild-Aufnahme für die Erkennung des QR-Codes feststellen, betrachten sie eben diesen Versuch als initiale Testphase. Jede Änderung der Umgebungsverhältnisse mit Auswirkung

auf die Qualität der Aufnahme interpretieren die Teilnehmer dann als zusätzliche Testphase, bis hin zu einer erfolgreichen Erkennung des QR-Codes.

Sollte sich die Firmware des Monocle-Systems nicht nach dem Anschalten des Geräts aktualisieren, kann die aktuelle Version des Monocle mit einem verbundenen Gerät abgefragt werden und mit der neusten angegebenen Version aus der offiziellen Dokumentation des Monocle auf der Seite des Herstellers BrilliantLabs abgeglichen werden. In diesem Falle suchen die Teilnehmer Hilfe im offiziellen Forum und starten das Gerät gegebenenfalls neu. Auch das Testen dieser Funktionalität betrachten die Teilnehmer als trivialen Black-Box-Test, dessen Ergebnis entweder der Erfolg oder das Scheitern des Update-Vorgangs aus unbekannten Gründen ist. Die Auftragnehmer bauen dann dynamisch weitere Testfälle ein, wie ein Neustarten der Internet-Verbindung oder des Gerätes.

2.5 Testumgebung

Die meisten der im Vorgehen beschriebenen Funktionen lassen sich direkt am Monocle-Gerät testen.

Das Einlesen des QR-Codes und die Funktionalität der Kamera lassen anhand das an dem Host gesendetem Laptop überprüfen. Zum Zeitpunkt der Abnahme ist dies also auf jedem möglichen Gerät mit integriertem Dateisystem, wie etwa Windows oder Android möglich. Zu diesem Zeitpunkt soll auch ein Smartphone als verbundenes Gerät möglich sein, um es für den Kunden einfach zu halten, das Monocle immer in der Nähe des Geräts zu behalten.

Zum testen der Verbindung per BLE-Kommunikation und des Programmcodes lässt sich während der Entwicklung und zum Zeitpunkt der Abnahme Visual Studio Code verwenden, unter Verwendung der für das Monocle vorgesehenen Erweiterung. Dieses Vorgehen ermöglicht den Teilnehmern und dem Kunden das Auslesen von Fehlermeldungen direkt über das verbundene Gerät. Für die Handy-App, die für die Abnahme verfügbar sein soll, um eine einfachere Verbindung mit dem Monocle-Gerät herzustellen, ist keine Anzeige für Fehlermeldungen vorgesehen.

Die Verschlüsselung mittels MicroPython-Skripten lässt sich auf sämtlicher Hardware mit Kompatibilität für MicroPython testen. Bei den Auftragnehmern kommt auch hier bei der Abnahme Visual Studio Code zum Einsatz.

3 Abnahmetest

Im nachfolgenden Kapitel wird die Abnahme des Produkts durch den Kunden beschrieben. Zunächst werden die zu prüfenden Anforderungen sowie die daraus abgeleiteten Testfälle dargelegt. Diese sind entscheidend für die Sicherstellung der Produktvollständigkeit bei der Übergabe. Der Abnahmetest dient als Beleg für die Erfüllung der zwischen Kunde und Auftragnehmer vereinbarten Leistungen. In den folgenden Seiten wird detailliert erörtert, wie die korrekte Funktionsweise des Programms überprüft und die Erfüllung der Anforderungen sichergestellt wird. Zudem wird erläutert, welche Testmethoden eingesetzt werden, um die angestrebten Ergebnisse zu erzielen. Anschließend wird auf die verwendeten Testskripte eingegangen und die Testfälle im Detail beschrieben.

3.1 Zu testende Anforderungen

Nr	Anforderung	Testfälle	Kommentar
1	F10 Ein- und Ausschaltfunktion des Monocle	T100	War das Ein- und Ausschalten des Monocle erfolgreich?
2	F20 Kommunikation zwischen dem Endgerät und dem Monocle	T200	War die Verbindung zwischen dem Endgerät und dem Monocle erfolgreich?
3	F30 Testen des Touchpads	T300	Funktioniert das Touchpad, wenn man es berührt?
4	F40 Lesen von QR-Codes	T400	Erkennt das Monocle den QR-Code?
5	F50 Entschlüsselung von Nachrichten	T500, T600	Kann man die versteckte Nachricht lesen oder ist sie verschlüsselt?
6	F60 Anzeigen der Nachrichten	T700	Zeigt das Monocle die entsprechende Nachricht?
7	F70 Funktionalität der Kamera	T800	Kann die Kamera Fotos aufnehmen?

3.2 Testverfahren

Das Testverfahren basiert auf dem Ansatz des spezifikationsbasierten Black-Box-Tests, bei dem keine Kenntnisse über die interne Struktur oder den Quellcode des zu testenden Systems erforderlich sind. Die Testausführung erfolgt basierend auf Eingabewerten, und die Komponenten werden daraufhin überprüft, ob sie zu diesen Eingaben passende Ausgaben liefern. Auf Grundlage der vorab definierten Modelle und Anforderungsdokumentationen lassen sich Testfälle leicht entwickeln und die Testergebnisse effektiv auswerten.

3.2.1 Testskripte

Es werden keine Testskripte verwendet.

3.3 Testfälle

3.3.1 Testfall $\langle T100 \rangle$ - Power On/Off

Ziel

Sicherstellen, dass das Monocle beim Herausnehmen aus der Gehäuse automatisch eingeschaltet wird und beim Zurücklegen ausschaltet.

Objekte/Methoden/Funktionen

Sensor für das Herausnehmen und Einlegen in das Gehäuse.

Pass/Fail Kriterien

Das Monocle schaltet sich beim Herausnehmen automatisch ein und beim Einlegen automatisch aus.

Vorbedingung

Das Monocle ist korrekt in das Gehäuse eingelegt, und das Gehäuse funktioniert fehlerfrei.

Einzelschritte

1. Nehmen das Monocle aus dem Gehäuse heraus.
2. Überprüfen, ob das Display aktiviert wird, indem man auf das Display schaut; man sollte das Wort 'Monocle' sehen.
3. Legen das Monocle zurück in das Gehäuse.

Beobachtungen / Log / Umgebung

Erfassen, ob es Fehlfunktionen beim Ein- oder Ausschalten gibt.

Besonderheiten

Notieren, ob das Monocle beim Ein- und Ausschalten visuelle oder akustische Signale gibt, die für die Bestätigung des Zustands wichtig sind.

Abhängigkeiten

Die Batterieladung des Monocle sollte ausreichend sein, wenn es aus dem Gehäuse genommen wird.

3.3.2 Testfall $\langle T200 \rangle$ - Bluetooth-Konnektivität

Ziel

Sicherstellung der stabilen Bluetooth-Verbindung zwischen Monocle und einem Host-Gerät.

Objekte/Methoden/Funktionen

Bluetooth-Verbindungs Aufbau.

Pass/Fail Kriterien

Der Test ist erfolgreich (Pass), wenn eine das Monocle-Display 'Waiting' anzeigt. Der Test ist nicht erfolgreich (Fail), wenn die Verbindung nicht stabil und konstant ist und das Display 'Waiting' nicht anzeigt.

Vorbedingung

Das Monocle muss aus seinem Gehäuse entfernt sein und liegt in der Nähe vom Host-Gerät.

Einzelschritte

1. Man bringt das Monocle in der Nähe vom Host-Gerät.
2. Der Aktivierungscode muss auf dem Host-Gerät ausgeführt werden, um die Bluetooth-Verbindung zu ermöglichen.

Beobachtungen / Log / Umgebung

Dokumentiere die Stärke und Qualität des Bluetooth-Signals während des Tests.

Besonderheiten

Führe den Test in verschiedenen Umgebungen durch (z.B. in einem Raum mit vielen anderen aktiven Bluetooth-Geräten).

Abhängigkeiten

Die Qualität der Bluetooth-Verbindung hängt von der Entfernung zwischen den verbundenen Geräten ab.

3.3.3 Testfall $\langle T300 \rangle$ - Touchpad-Reaktivitäts

Ziel

Überprüfen der Funktionalität und Reaktionsfähigkeit des Touchpads.

Objekte/Methoden/Funktionen

Berührungseingaben auf dem Touchpad.

Pass/Fail Kriterien

Der Test ist erfolgreich (Pass), wenn das Monocle-Display 'Pad A touched' oder 'Pad B touched' zeigt und nicht erfolgreich (Fail), wenn das Display 'Touch a Pad' anzeigt.

Vorbedingung

Monocle ist eingeschaltet und mit dem Host-Gerät verbunden sein.

Einzelschritte

1. Führen den spezifischen Code auf dem Host-Gerät aus, um die Interaktion zu starten.
2. Das Display des Monocle zeigt die Aufforderung „Touch a Pad“.
3. Berühren das Touchpad A oder B auf dem Monocle.
4. Je nachdem, welches Pad berührt wurde, zeigt das Display „Pad A touched“ oder „Pad B touched“.

Beobachtungen / Log / Umgebung

Genauigkeit und Zeitverzögerung der Eingaben.

Besonderheiten

Testen der Reaktion des Touchpads auf verschiedene Druckintensitäten.

Abhängigkeiten

Dieser Test hängt von der korrekten Funktion des Touchpad-Code-Detektors ab.

3.3.4 Testfall $\langle T400 \rangle$ - QR-Code-Erkennung

Ziel

Überprüfen der Fähigkeit des Monocle, verschiedene QR-Codes korrekt zu erkennen und zu verarbeiten.

Objekte/Methoden/Funktionen

Kamera des Monocle, QR-Code-Erkennungssoftware.

Pass/Fail Kriterien

Der Test ist erfolgreich (Pass), wenn das Monocle alle bereitgestellten QR-Codes korrekt erkennt und die darin enthaltenen Informationen richtig interpretiert. Ein Fehlschlagen (Fail) liegt vor, wenn QR-Codes nicht erkannt werden.

Vorbedingung

Das Monocle muss eingeschaltet und mit dem Host-Gerät verbunden sein.

Einzelschritte

1. Platzieren des QR-Codes in einer angemessenen Entfernung vor dem Monocle.
2. Berühren das Touchpads A oder B, um den Prozess zu starten.
3. Das Touchpad loslassen.
4. Das aufgenommene Foto wird analysiert, um einen QR-Code zu finden.
5. Wenn ein QR-Code gefunden wird, zeigt das Display die im QR-Code enthaltenen Informationen an, sei es ein Link oder ein Text.

Beobachtungen / Log / Umgebung

Dokumentation der Erkennungszeit, der Lichtverhältnisse während des Tests und aller Fehlermeldungen oder Probleme bei der Erkennung.

Besonderheiten

Testen unter verschiedenen Lichtbedingungen, um die QR-Code-Erkennung unter verschiedenen Umgebungsbedingungen zu bewerten.

Abhängigkeiten

Die Software für die QR-Code-Erkennung muss auf dem neuesten Stand sein. Die Funktionalität kann auch von der aktuellen Firmware-Version des Monocle abhängen.

3.3.5 Testfall $\langle T500 \rangle$ - Entschlüsselung von QR-Codes

Ziel

Überprüfung der korrekten Entschlüsselung von Nachrichten in QR-Codes, um die Sicherheit und Integrität der Datenübertragung zu gewährleisten.

Objekte/Methoden/Funktionen

Entschlüsselungsalgorithmus, Host.

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn die Nachricht nach dem Scan des QR-Codes korrekt entschlüsselt und angezeigt wird. Der Test ist fehlgeschlagen, wenn die Nachricht nicht entschlüsselt werden kann oder Fehler in der Darstellung auftreten.

Vorbedingung

Das Entschlüsselungssystem muss auf dem Host ordnungsgemäß konfiguriert und betriebsbereit sein.

Einzelschritte

1. Auswahl einer Nachricht für die Entschlüsselung.
2. Anwendung des Entschlüsselungsalgorithmus
3. Vergleich der entschlüsselten Nachricht mit der Originalnachricht.

Beobachtungen / Log / Umgebung

Dokumentation aller technischen Schwierigkeiten oder Abweichungen während des Tests.

Besonderheiten

Beachtung der Leistungsfähigkeit des Entschlüsselungsalgorithmus.

Abhängigkeiten

Dieser Test ist abhängig von der korrekten Funktion des Entschlüsselungsalgorithmus.

3.3.6 Testfall $\langle T600 \rangle$ - Erkennung verschlüsselter QR-Codes

Ziel

Überprüfung, ob das Monocle in der Lage ist, verschlüsselte QR-Codes von unverschlüsselten zu unterscheiden und als verschlüsselt zu kennzeichnen.

Objekte/Methoden/Funktionen

QR-Code-Erkennungs-Algorithmus, Verschlüsselungserkennungsalgorithmus.

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn das Monocle-Gerät alle getesteten verschlüsselten QR-Codes korrekt als verschlüsselt identifiziert und unverschlüsselte QR-Codes entsprechend als unverschlüsselt erkennt. Der Test ist fehlgeschlagen, wenn das Gerät einen oder mehrere QR-Codes falsch klassifiziert.

Vorbedingung

Mehrere Test-QR-Codes, sowohl verschlüsselte als auch unverschlüsselte, sind vorbereitet und zugänglich.

Einzelschritte

1. Starten QR-Code-Scanner-Anwendung.
2. Nacheinander Scannen der bereitgestellten QR-Codes.
3. Jeder gescannte QR-Code wird durch den Verschlüsselungserkennungsalgorithmus analysiert.
4. Überprüfung, ob die Ausgabe des Systems jeden QR-Code korrekt als verschlüsselt oder unverschlüsselt kategorisiert.

Beobachtungen / Log / Umgebung

Protokollierung der Ergebnisse jeder QR-Code-Erkennung und Analyse von Fehlalarmen oder Fehlklassifikationen.

Besonderheiten

Besondere Aufmerksamkeit gilt der Erkennungsgenauigkeit bei QR-Codes, die mit ähnlichen, aber unterschiedlichen Verschlüsselungstechniken erstellt wurden.

Abhängigkeiten

Dieser Test ist abhängig von der korrekten Funktion und Kalibrierung des QR-Code-Scanners sowie der Effektivität des Verschlüsselungserkennungsalgorithmus.

3.3.7 Testfall $\langle T700 \rangle$ - Anzeigen der Nachricht

Ziel

Dieser Testfall überprüft, ob das Monocle in der Lage ist, eine verschlüsselte Nachricht nach der Entschlüsselung korrekt anzuzeigen.

Objekte/Methoden/Funktionen

Monocle-Display, Nachricht, Anzeigefunktion des Monocle-Geräts.

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn das Monocle die entschlüsselte Nachricht korrekt und vollständig anzeigt. Ein Fehlschlag tritt ein, wenn die Nachricht unvollständig, fehlerhaft oder gar nicht angezeigt wird.

Vorbedingung

Eine verschlüsselte Nachricht ist auf dem Host und das Monocle-Gerät ist eingeschaltet und funktionsbereit.

Einzelschritte

1. Start des Monocle
2. Auswahl der verschlüsselten Nachricht zur Anzeige.
3. Überprüfung, ob die Nachricht nach der Entschlüsselung auf dem Display erscheint.
4. Vergleich der angezeigten Nachricht mit der erwarteten Entschlüsselung.

Beobachtungen / Log / Umgebung

Erfassung aller Systemantworten und etwaiger Fehlermeldungen während des Tests.

Besonderheiten

Überprüfung der Reaktionszeit des Systems und der Lesbarkeit der angezeigten Nachricht unter verschiedenen Lichtbedingungen.

Abhängigkeiten

Dieser Test hängt von der korrekten Funktion des Entschlüsselungsalgorithmus und des Display ab.

3.3.8 Testfall $\langle T800 \rangle$ - Funktionalität der Kamera

Ziel

Überprüfung der Funktionalität der Kamera im Monocle, insbesondere ihre Fähigkeit, klare und präzise Fotos aufzunehmen.

Objekte/Methoden/Funktionen

Monocle, Bildverarbeitungssoftware, Übertragungsfunktion zum verbundenen Host, QR-Code Lesegerät

Pass/Fail Kriterien

Der Test ist erfolgreich, wenn die Kamera Fotos entsprechend den Spezifikationen von QR-Codes aufnimmt. Der Test ist fehlgeschlagen, wenn die Fotos nicht Scharf genug sind um die QR-Codes zu erkennen.

Vorbedingung

Das Monocle-Gerät ist eingeschaltet, die Kamera ist aktiviert, und ein Test QR-Code ist positioniert.

Einzelschritte

1. Aktivieren der Kamerafunktion auf dem Monocle.
2. Ausrichten der Kamera auf das Testobjekt und Aufnahme eines Fotos.
3. Überprüfung der Bildqualität auf dem verbundenen Computer.
4. Überprüfung der Lesbarkeit des QR-Codes mit einem Lesegerät.
5. Wiederholung des Tests unter verschiedenen Lichtbedingungen und Entfernungen.

Beobachtungen / Log / Umgebung

Dokumentation der Bildqualität und aller technischen Probleme während des Tests.

Besonderheiten

Beachtung der Auswirkungen von unterschiedlichen Umgebungslichtverhältnissen auf die Bildqualität.

Abhängigkeiten

Dieser Testfall ist abhängig von der korrekten Funktion der Kamera und des QR-Code Lesegeräts.

4 Integrationstest

Im folgenden Kapitel wird die genaue Vorgehensweise des Integrationstests beschrieben. Zunächst wird das Testziel genannt, dann werden die zu testenden Komponenten, das Testverfahren, die Testskripte und zuletzt die konkreten Testfälle erörtert. Ähnlich wie im Abnahmetestspezifikation des Abnahmetests soll die Bearbeitung erfolgen.

Testziel des Integrationstest ist eine reibungslose Zusammenarbeit der einzelnen Komponenten. Es wird also geschaut ob die Kommunikation zwischen den Komponenten funktioniert.

4.1 Zu testende Komponenten

Nr	Komponenten	Testfälle	Kommentar
1	<C00> FPGA	<T200>	
2	<C10> Host-Anwendung	<T200>	
3	<C20> QR-Code-Erkennung	<T200>, <T210>	
4	<C30> Entschlüsselung-Programm	<T200>, <T210>	
5	<C40> Monocle-Anwendung	<T200>	
6	<C50> Bluetooth-MCU	<T200>	
7	<C60> Kamera	<T200>, <T220>	
8	<C70> Toypad	<T200>, <T220>	
9	<C80> Display	<T200>	

4.2 Testverfahren

Um die Integration der Komponenten zu überprüfen, werden Black-Box-Tests eingesetzt. Diese stellen sicher, dass die Software gemäß den definierten funktionalen Anforderungen korrekt funktioniert. Die Komponente <C00>, <C50>, <C60>, <C70> und <C80> wurden bereits von dem Hersteller des Monocles bereitgestellt. Daher ist die Integration dieser Komponente nicht etwas, was in unseren Testfall berücksichtigt wird.

4.2.1 Testskripte

Da die Black-Box-Tests bereits die korrekte Funktionalität der Software sicherstellen, werden im Integrationstest keine Testskripte benötigt.

4.3 Testfälle

Nach den erfolgreichen Unit-Tests innerhalb der einzelnen Komponenten wird nun im Folgenden die Zusammenarbeit zwischen den Komponenten genauer betrachtet.

4.3.1 Testfall $\langle T200 \rangle$ - Komponente C00+C10+C40+C50+C60+C70

Ziel Es soll getestet werden, ob die Monocle-Anwendung die Bilder richtig an die Host-Anwendung übergibt.

Objekte/Methoden/Funktionen

main()

Pass/Fail Kriterien

Pass: Der Test ist erfolgreich wenn das eingelesene Bild erfolgreich an den Host gesendet wird.

Fail: Der Test schlägt fehl, wenn das eingelesene Bild nicht an den Host gesendet wird.

Vorbedingung

Das Monocle ist in der Reichweite vom Host, so dass eine Bluetooth Verbindung bestehen kann.

Einzelschritte

Eingabe:

1. Host-Anwendung öffnen und Monocle anschalten
2. Monocle auf das einzulesene Bild richten und Touchpad bedienen
3. Warten bis das Monocle das Bild einliest und übermittelt

Ausgabe:

Bild ist auf Host zu sehen.

Beobachtungen / Log / Umgebung Darauf achten, ob das richtige Bild an den Host gesendet wurde

Besonderheiten -

Abhängigkeiten

$\langle T100 \rangle$, $\langle T200 \rangle$, $\langle T300 \rangle$, $\langle T800 \rangle$

4.3.2 Testfall $\langle T210 \rangle$ - Komponente C20 + C30

Ziel Testen, ob die Botschaft des QR-Codes richtig entschlüsselt wird

Objekte/Methoden/Funktionen

get_key()

decrypt_message()

Pass/Fail Kriterien

Pass: Der Test ist erfolgreich, wenn die Botschaft korrekt entschlüsselt wird.

Fail: Der Test schlägt fehl, wenn die Botschaft nicht korrekt entschlüsselt wird.

Vorbedingung

Der korrekte Schlüssel für die Entschlüsselung wird verwendet.

Einzelschritte

Eingabe:

1. Die Verschlüsselte Nachricht, sowie der Key muss an den Code zuständig für die Entschlüsselung übergeben werden.

Ausgabe:

2. Das Programm wird ausgeführt und returnt in der Konsole die entschlüsselte Nachricht

Beobachtungen / Log / Umgebung

Darauf achten, ob die Nachricht korrekt aus dem QR-Code ausgelesen wird.

Besonderheiten

Die verschlüsselten Nachrichten und Schlüssel sind 'hard-gecoded'

Abhängigkeiten

$\langle T500 \rangle$, $\langle T600 \rangle$

4.3.3 Testfall $\langle T220 \rangle$ - Komponente C60 + C70

Ziel Testen, ob bei Bedienung des Touchpads, ein Bild aufgenommen wird.

Objekte/Methoden/Funktionen

`trigger_capture()`

Pass/Fail Kriterien

Pass: Der Test ist erfolgreich, wenn bei der Bedienung des Touchpads, die Kamera das Bild aufnimmt.

Fail: Der Test schlägt fehl, wenn durch die Bedienung des Touchpads die Kamera nicht funktioniert und das Bild nicht aufnimmt.

Vorbedingung

-

Einzelschritte

Man positioniert das Monocle korrekt zum Bild und betätigt das Touchpad

Beobachtungen / Log / Umgebung

-

Besonderheiten

-

Abhängigkeiten

$\langle T100 \rangle$, $\langle T800 \rangle$

5 Unit-Tests

In diesem Kapitel wird die Vorgehensweise für die Unit-Tests beschrieben. Die Bearbeitung erfolgt analog zur Abnahmetestspezifikation. Es ist darauf zu achten, dass die Nummerierung der Testfälle sich nicht mit der aus der Abnahmetestspezifikation überschneidet.

Testziel: Das Ziel der Unit-Tests ist es, einzelne Softwarekomponenten isoliert zu testen, um sicherzustellen, dass jede Komponente korrekt funktioniert.

5.1 Zu testende Komponenten

Nr	Komponenten	Testfälle	Kommentar
1	<F10> Ein- und Ausschaltfunktion des Monocle	<T300>	Testet, ob das Monocle erfolgreich ein- und ausgeschaltet werden kann.
2	<F20> Kommunikation zwischen dem Endgerät und dem Monocle	<T301>	Testet die Verbindung zwischen dem Endgerät und dem Monocle.
3	<F30> Touchpad	<T302>	Testet, ob das Touchpad auf Berührungen reagiert.
4	<F40> QR-Code Lesen	<T303>	Testet, ob das Monocle QR-Codes korrekt lesen kann.
5	<F50> Entschlüsselung von Nachrichten	<T304,T305>	Testet die Fähigkeit zur Entschlüsselung von Nachrichten.
6	<F60> Anzeigen der Nachrichten	<T306>	Testet, ob Nachrichten korrekt auf dem Monocle angezeigt werden.
7	<F70> Kamera	<T307>	Testet die Funktionalität der Kamera.

5.2 Testverfahren

Die detaillierte Vorgehensweise zur Durchführung der Unit-Tests wird hier beschrieben, um eine einheitliche Durchführung der Tests zu gewährleisten.

5.2.1 Testskripte

Die Testskripte sind ein zentraler Bestandteil des Testverfahrens. Hier werden die Skripte dokumentiert, die zur Automatisierung der Tests verwendet werden.

5.3 Testfälle

5.3.1 Testfall $\langle T300 \rangle$ - Ein -und Ausschaltfunktion des Monocle

Ziel Überprüfung, ob das Monocle richtig ein -und ausgeschaltet werden kann.

Objekte/Methoden/Funktionen Monocle.___aenter___(), Monocle.___aexit___()

Pass/Fail Kriterien Der Test ist bestanden, wenn das Monocle nach dem Einschalten betriebsbereit ist und nach dem Ausschalten vollständig heruntergefahren ist.

Vorbedingung Das Monocle ist aus dem Gehäuse entfernt.

Einzelschritte

1. Monocle aus dem Gehäuse herausnehmen
2. Zustand prüfen
3. Monocle ausschalten

Beobachtungen / Log / Umgebung Keine

Besonderheiten Keine

Abhängigkeiten Keine

5.3.2 Testfall $\langle T301 \rangle$ - Kommunikation zwischen dem Endgerät und dem Monocle

Ziel Überprüfung, ob die Kommunikation zwischen dem Endgerät und dem Monocle erfolgreich ist.

Objekte/Methoden/Funktionen `Monocle.connect()`, `Monocle.send_command()`

Pass/Fail Kriterien Der Test ist bestanden, wenn die Verbindung erfolgreich ist.

Vorbedingung Endgerät und Monocle sind eingeschaltet.

Einzelschritte

1. Verbindung herstellen
2. Kommunikation testen (ein Befehl senden und auf den Antwort warten)

Beobachtungen / Log / Umgebung Keine

Besonderheiten Keine

Abhängigkeiten Testfall T300 muss bestanden sein.

5.3.3 Testfall $\langle T302 \rangle$ - Touchpad

Ziel Überprüfung, ob das Touchpad auf Berührungen reagiert.

Objekte/Methoden/Funktionen `trigger_capture(button)`

Pass/Fail Kriterien Der Test ist bestanden, wenn Berührungen korrekt erkannt.

Vorbedingung Das Monocle ist eingeschaltet und mit dem Endgerät verbunden.

Einzelschritte

1. Touchpad berühren
2. Reaktion prüfen

Beobachtungen / Log / Umgebung Keine

Besonderheiten Keine

Abhängigkeiten Testfälle T300/T301 müssen bestanden sein.

5.3.4 Testfall $\langle T303 \rangle$ - QR-Code Lesen

Ziel Überprüfung, ob das Monocle QR-Codes korrekt erkennen und lesen kann.

Objekte/Methoden/Funktionen detect()

Pass/Fail Kriterien Der Test ist bestanden, wenn QR-Codes korrekt erkannt und gelesen wird.

Vorbedingung Das Monocle ist eingeschaltet und bereit, QR-Codes zu scannen.

Einzelschritte

1. QR-Code vor der Kamera halten
2. QR-Code lesen
3. Überprüfung des gelesenen Codes

Beobachtungen / Log / Umgebung Ausgabe der QR-Code-Daten im Terminal sowie auch auf das Display.

Besonderheiten Keine

Abhängigkeiten Testfälle T300/T301 müssen bestanden sein.

5.3.5 Testfall $\langle T304 \rangle$ - Entschlüsselung von QR-Codes

Ziel Überprüfung der korrekten Entschlüsselung von Nachrichten in QR-Codes.

Objekte/Methoden/Funktionen `get_key()` und `decrypt_message()`

Pass/Fail Kriterien Der Test ist erfolgreich, wenn die Nachricht korrekt entschlüsselt wird und lesbar ist.

Vorbedingung

1. Das Entschlüsselungssystem muss auf dem Host ordnungsgemäß konfiguriert sein.
2. Ein QR-Code mit einer verschlüsselten Nachricht.

Einzelschritte

1. Den Entschlüsselungsalgorithmus aktivieren.
2. Die Funktion `get_key()` liefert den verwendeten Schlüssel.
3. Der Schlüssel wird an die Funktion `decrypt_message()` übergeben.
4. Die Funktion `decrypt_message()` gibt die entschlüsselte Nachricht zurück.

Beobachtungen / Log / Umgebung Im Terminal soll die ursprüngliche Nachricht angezeigt werden.

Besonderheiten Keine

Abhängigkeiten Der Entschlüsselungsprozess soll auf demselben Host durchgeführt werden, auf dem die Verschlüsselung erfolgte.

5.3.6 Testfall $\langle T305 \rangle$ - Erkennung verschlüsselter QR-Codes

Ziel Überprüfung, ob das Monocle in der Lage ist, verschlüsselte QR-Codes von unverschlüsselten zu unterscheiden.

Objekte/Methoden/Funktionen detect()

Pass/Fail Kriterien Der Test ist bestanden, wenn den verschlüsselten QR-Codes korrekt erkannt und gelesen wird.

Vorbedingung Das Monocle ist eingeschaltet und bereit, QR-Codes zu scannen.

Einzelschritte

1. verschlüsselter QR-Code vor der Kamera halten
2. verschlüsselter QR-Code lesen
3. Überprüfung des gelesenen Codes, und ob der verschlüsselter QR-Code korrekt entschlüsselt.

Beobachtungen / Log / Umgebung Ausgabe der verschlüsselten QR-Code-Daten im Terminal sowie auch auf das Display.

Besonderheiten Keine

Abhängigkeiten Testfälle T300/T301/T302/T303/T304 müssen bestanden sein.

5.3.7 Testfall $\langle T306 \rangle$ - Anzeigen der Nachrichten

Ziel Überprüfung, ob Nachrichten korrekt auf dem Monocle angezeigt werden.

Objekte/Methoden/Funktionen `display()`

Pass/Fail Kriterien Der Test ist bestanden, wenn Nachrichten korrekt angezeigt werden.

Vorbedingung Das Monocle ist eingeschaltet und mit dem Endgerät verbunden.

Einzelschritte

1. QR-Code scannen
2. QR-Code erkannt und entschlüsselt
3. Anzeigen der Nachricht

Beobachtungen / Log / Umgebung Ausgabe der Nachricht im Display.

Besonderheiten Keine

Abhängigkeiten Testfälle T300/T301/T302 müssen bestanden sein.

5.3.8 Testfall $\langle T307 \rangle$ - Kamera

Ziel Überprüfung, ob das Kamera vom Monocle ein Bild aufnimmt.

Objekte/Methoden/Funktionen `camera.capture()`

Pass/Fail Kriterien Der Test ist bestanden, wenn das Bild aufgenommen wird und auf das Endgerät gespeichert wurde.

Vorbedingung Das Monocle ist eingeschaltet, mit dem Endgerät verbunden und ein Touchpad ist berührt.

Einzelschritte

1. Touchpad berühren
2. Bild aufnehmen
3. Bild auf das Endgerät gespeichert

Beobachtungen / Log / Umgebung Bild aufgenommen und auf Endgerät gespeichert.

Besonderheiten Keine

Abhängigkeiten Testfälle T300/T301/T302 müssen bestanden sein.

6 Glossar

(Alphabetisch und absteigend sortiert)

AES - Steht für Advanced Encryption Standard, welches eine Blockchiffre ist, die zum Ver- und Entschlüsseln von Informationen verwendet wird.

BLE-Kommunikation - Bluetooth Low Energy-Kommunikation ist eine Technik zur drahtlosen Vernetzung von Hardware, die auf eine Reichweite von unter 10 Metern ausgelegt ist.

Empfangscharakteristik - Bezieht sich auf die spezifischen Eigenschaften eines Systems oder einer Komponente, die bestimmen, wie Daten empfangen werden.

IAS - Institut für Anwendungssicherheit an der TU Braunschweig.

Komponentendiagramm - UML-Diagramm zur Kenntlichmachung der logischen Komponenten einer Software, sowie dessen für andere Komponenten zur Verfügung gestellten Schnittstellen.

MicroPython - ist eine Implementierung der Programmiersprache Python, die speziell für Mikrocontroller und eingebettete Systeme entwickelt wurde.

QR-Code - Kurz für Quick Response, ist ein zweidimensionaler Strichcode, welcher Informationen in Form von Zeichen und Ziffern speichert.

UART - Steht für Universal Asynchronous Receiver-Transmitter.

UUID - Steht für Universally Unique Identifier (universell eindeutiger Bezeichner).

Übertragungscharakteristik - Bezieht sich auf die spezifischen Eigenschaften eines Systems oder einer Komponente, die bestimmen, wie Daten übertragen werden.

Zustandsdiagramm - UML-Diagramm, das sich auf die Zustände eines Produkts oder dessen Software konzentriert, sowie deren Übergänge und Subzustände.