



SOFTWARE- ENTWICKLUNGSPRAKTIKUM (SEP)

ARID - AUGMENTED REALITY IN DISGUISE

Software-Entwicklungspraktikum (SEP)
Sommersemester 2024

Abnahmetestspezifikation

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, 28. Mai 2024

Bearbeiterübersicht

| Kapitel | Autoren | Kommentare |
|---------|-------------------|---------------------|
| 1 | Vyvy, Amir | — |
| 2 | Mohamed Ali, Vyvy | — |
| 2.1 | Mohamed Ali | — |
| 2.2 | Mohamed Ali | — |
| 2.3 | Mohamed Ali | — |
| 2.4 | Jadon | — |
| 2.5 | Jadon | — |
| 3 | Ibrahim, Tim | — |
| 3.1 | Ibrahim | — |
| 3.2 | Tim | — |
| 3.3 | Ibrahim, Tim | — |
| 4 | Sämtliche | Fortlaufend ergänzt |

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 5 |
| 2 | Testplan | 7 |
| 2.1 | Zu testende Komponenten | 7 |
| 2.2 | Zu testende Funktionen/Merkmale | 7 |
| 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 | Glossar | 22 |

Abbildungsverzeichnis

1 Einleitung

In der Abnahmetestspezifikation wird die Software anhand von Test untersucht. Das System wird während und nach der Entwicklung getestet, um eine bestmögliche Qualität der Software sicherzustellen.

In dem Projekt Augmented Reality In Disguise ist das gründliche Testen unerlässlich, da hier mit vertraulichen Daten umgegangen wird. Ein Teil ist das Entwickeln des Steganogramms, bei dem ein QR-Code mit einer geheimen verschlüsselten Nachricht in einem Bild integriert wird. Zu offensichtlich erkennbare QR-Codes, würden dazu führen, dass herauskommt, dass kommuniziert wird und die Vertraulichkeit verletzt wird. Zudem müssen auch die korrekte Ver- und Entschlüsselung der Nachricht funktionieren, da es sonst zum Leak der Nachricht kommen könnte oder es auf der User Seite selbst nicht mehr möglich wäre, die Nachricht zu entschlüsseln. Das Monocle, welches zum Lesen des QR-Codes verwendet wird, muss den QR-Code selbstverständlich fehlerfrei erkennen können.

Das Projekt wird auf ihre einzelnen Funktionen getestet, die Tests und deren Ablauf werden im Abnahmetestspezifikation dokumentiert. Zu testende Funktionen sind unter anderem:

- Bild mit korrekt integriertem QR-Code generieren: Ist einer der wichtigsten Funktionen, da der QR-Code nicht unlesbar oder zu leicht erkennbar sein darf. Dies würde dazu führen, dass es zu einfach zu erkennen ist, dass eine Botschaft in den QR-Code versteckt ist.
- verstecktes QR-Code wird vom Monocle erkannt: Diese Funktion ist unerlässlich. Dabei ist es wichtig das der QR-Code fehlerfrei, schnell und zuverlässig erkannt wird.
- verschlüsseltes QR wird vom Monocle gelesen: Nachdem der QR-Code erkannt wird, ist es wichtig das dieser auch richtig ausgelesen wird. Eine fehlerhafte Auslesung des QR-Codes wäre fatal, da dann der Sinn und Zweck des Produktes nicht erfüllt wird. Die Anforderung dabei ist das schnelle und korrekte Auslesen der Botschaft.
- korrekte Entschlüsselung der Botschaft: Da die Botschaften verschlüsselt sind, sollen diese auch richtig entschlüsselt werden. Der Träger des Monocle soll also die ihm entsprechende Nachricht erhalten. Die Botschaft soll ohne Verlust an Information und Daten entschlüsselt werden, dies soll reibungslos und fehlerfrei stattfinden.

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 folgenden Kapitel wird der Testplan beschrieben. Dabei ist das Hauptmerkmal die Qualitätssicherung, darum muss diese frühzeitig ausgearbeitet werden. Im kommenden Abschnitt wird die Vorgehensweise und der Umfang dargestellt. Die zu testenden Komponenten und ihre Eigenschaften und Funktionen werden aufgelistet, außerdem werden explizit die nicht zu testenden Funktionen auch identifiziert.

2.1 Zu testende Komponenten

Im folgenden Abschnitt werden die Komponenten aufgelistet, die im Rahmen der Tests berücksichtigt werden (sehen Sie auch Pflichtenheft Abschnitt 8):

- **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.
- **Bluetooth-MCU** : Die Haupt-CPU vom Monocle ist die Bluetooth-MCU. Der Nordic nRF52832-Speicher ist die verwendete MCU, die Bluetooth 5.2 unterstützt.
- **Display** : (Siehe Pflichtenheft, Abschnitt 8.2)
- **Kamera** : Die Omnivision OV5640 Frontkamera des Monocle ist direkt mit dem FPGA verbunden.
- **Touchpads** : Ein Azoteq IQS620A Touch Controller empfängt Eingaben von den Touchpads A und B oben auf dem Monocle-Gerät.

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 wie gut die Qualität des QR-Code ab. Deswegen wird dies stark nicht getestet.
- **Ladezeit des Codes** : Es gibt keine festen Kriterien für akzeptable Ladezeiten, da diese variieren können.
- **Micro-/Python Libraries** : Diese Bibliotheken werden in der Regel als vertrauenswürdig und funktional betrachtet, deswegen werden wir die nicht explizit testen.
- **Kommunikationsgeschwindigkeit durch BLE** : Die Geschwindigkeit der Kommunikation durch Bluetooth Low Energy (BLE) wird nicht spezifisch getestet, da es an viele Umgebungsbedingungen liegt, und davon sind Signalstärke, wie weit von dem Endgerät kann man das Monocle nutzen, und der Implementierung der BLE-Protokolle.
- **Ladezeit des Monocle** : Da ist es für die Benutzererfahrung als unwichtig ertrachtet 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 Erfolgschance 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 wird 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, finden die Teilnehmer dies heraus, indem sie die aktuelle Version des Monocle von einem verbundenen Gerät aus abfragen und mit der neusten angegebenen Version in der offiziellen Dokumentation des Monocle auf der Seite des Herstellers BrilliantLabs abgleichen. 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 Funktionalität der Kamera lassen sich anhand des an den Laptop gesendeten Bilds prüfen. 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 Abschnitt 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.

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 Glossar

(Alphabetisch und absteigend sortiert)

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

IEEE 829-Standard - Softwaretest Standard, IEEE steht Institute of Electrical and Electronics Engineers.

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.

Steganogramm - Trägermedium, welches durch die Anwendung von Steganographie eine Nachricht enthält.

Steganographie - Wissenschaft der verborgenen Speicherung von Informationen in einem Trägermedium.

Visual Code Studio - Ist eine integrierte Entwicklungsumgebung (IDE) von Microsoft, die eine Vielzahl von Tools und Funktionen für die Softwareentwicklung bietet.