

Hochschule für Technik und Wirtschaft Berlin
Masterstudiengang
Angewandte Informatik

Bedienung und Verwaltung von Haushaltsgeräten mittels
NFC am Beispiel eines Kaffeevollautomaten

Forschungsprojekt 1
Prof. Dr. Jürgen Sieck

von
Maurus Rohrer
0526055
30.09.2012

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziel	1
2	Grundlagen und Technologien	3
2.1	NFC	3
2.1.1	Read-Write Modus	3
2.1.2	Peer-to-Peer Modus	4
2.2	MDB	5
2.2.1	Funktionsweise	6
2.2.2	Protokoll Eigenschaften	6
2.2.3	Cashless Device	7
2.2.4	MDB2PC	8
3	Existierende Systeme für die Abrechnung und Verwaltung von Kaffeemaschinen	10
3.1	Externe Steuerung und Verwaltung	10
3.1.1	Jura Web Pilot	10
3.1.2	Textspresso von Zipwhip	11
3.2	Abrechnungssysteme	11
3.2.1	Jura MDB-Interface	11
3.2.2	Münz- und Notenautomat	12
3.2.3	Bargeldlose Systeme	13
3.3	Smart Saeco	13
4	System Anforderungen	15
4.1	Nichtfunktionale Anforderungen	15
4.1.1	Flexibilität	15
4.1.2	Kosten	15
4.1.3	Benutzbarkeit	16
4.2	Funktionale Anforderungen	16
4.2.1	Abrechnung	16
4.2.2	Verwaltung	17
4.2.3	Automatische Konfiguration und Bezug von Kaffee	18
5	Lösungsvorschläge	19
5.1	Lösungsvorschlag 1 - MDB	19
5.2	Lösungsvorschlag 2 - MDB, Robotik	20
5.3	Lösungsvorschlag 3 - MDB, Robotik, Kamera	22
5.4	Lösungsvorschlag 4 - MDB, HW Manipulation, Kamera	23
5.5	Lösungsvorschläge Bewertung und Entscheidung	23

6	Möglichkeiten der Bedienung	25
6.1	Interaktion mit dem System	25
6.2	Bereitstellung der Informationen	26
6.3	Registrierung	27
6.4	Änderung der Daten	28
7	Prototyp Implementierung	29
7.1	Systemarchitektur	29
7.2	Datenspeicherung	30
7.3	Kommunikationsprotokolle	30
	7.3.1 QT Signal & Slot	32
	7.3.2 D-Bus	32
7.4	NFC Modul	33
7.5	MDB Modul	34
7.6	Screen Caputre Modul	35
7.7	Web Kiosk	37
7.8	Webserver	38
8	Fazit	39
8.1	Ausblick	39

Abbildungsverzeichnis

1	NFC Forum Architektur und Betriebsmodi [14]	4
2	NFC Peer-to-Peer Modus Protokollstapel [14]	5
3	MDB Cashless Device Zustandsdiagramm	8
4	MDB2PC Interface Protokoll [8]	9
5	MDB2PC Interface [2]	9
6	Textpresso von Zipwhip [3]	12
7	Smart Saeco Hardware Modifikationen [10]	14
8	Lösungsvorschlag 1 - MDB	19
9	Lösungsvorschlag 2 - MDB, Robotik	21
10	Lösungsvorschlag 3 - MDB, Robotik, Kamera	22
11	Lösungsvorschlag 4 - MDB, HW Manipulation, Kamera	23
12	Prototyp Systemarchitektur	29
13	Datenbank Diagramm	31
14	QT Signal & Slot Konzept [13]	32
15	MDB Verkauf	35
16	Kamera Bild RGB	36
17	HSV transformiertes Bild	36
18	Threshold Filter	36
19	Gaussian smoothing Filter	37

Tabellenverzeichnis

1	NFC Forum unterstützte RFID-Transponder	4
2	Serial Bit Format MDB	6
3	MDB Cashless Device Betriebs- und Bezugsbereit Übersicht	8
4	Lösungsvorschläge Vergleich	24
5	Interaktionsmöglichkeiten Vergleich	26

Listings

1	NFC-RFID Bibliothek Transponder-ID lesen	33
2	Serielles MDB-Frame	34
3	Word Matching Algorithmus	37

Abkürzungsverzeichnis

HSV	Hue, Saturation and Value
HTTP	Hypertext Transfer Protocol
IPC	Inter-Process Communication
MDB	Multi Drop Bus
MVC	Model View Controller
NFC	Near Field Communication
ORB	Object Request Broker
RFID	Radio-Frequency Identification
RGB	Rot, Grün und Blau
SMS	Short Message Service
SQL	Structured Query Language
VMC	Vending Mechanism Controller

1 Einleitung

Ziel dieses Forschungsprojekt ist es, flexible und kostengünstige Varianten für die Verwaltung und Bedienung von Kaffeevollautomaten mittels NFC zu erforschen. Zudem werden die Möglichkeiten der NFC Technologie im Kontext der Bedienung und der Verwaltung von Haushaltsgeräten analysiert und demonstriert.

Handelsübliche Verwaltungssystemen von Kaffeevollautomaten realisieren meist nur den Abrechnungsprozess und sind teuer in der Beschaffung. Es existieren noch keine Systeme welche den Abrechnungsprozess, die Verwaltung sowie die automatische Konfiguration von Kaffeerezepten für unterschiedliche Modelle und Hersteller von Kaffeemaschinen in einem System realisieren. Im Rahmen dieser Arbeit, werden die Grundlagen sowie verschiedenen Möglichkeiten für die Realisierung eines solchen umfänglichen Systems erforscht.

1.1 Motivation

In kleineren Unternehmen oder Gemeinschaften, welche im besitz eines Kaffeevollautomaten sind, kommt es häufig und schneller als erwünscht zu den Diskussionen wer den nächsten Kaffee besorgen oder die Maschine reinigen soll. Ist ein klassisches Abrechnungssystem, bei welchem man immer über das nötige Wechselgeld verfügen muss nicht erwünscht, bleibt meist nur die von Hand geführte Strichliste. Diese ist jedoch nicht so zuverlässig wie ein elektronisches Abrechnungssystem. Ziel ist ein Abrechnungs- und Bediensystem zu entwerfen, welches kostengünstige Hardware verwendet und offene Software zur Verfügung stellt. Sodass interessierte Unternehmen die Möglichkeit haben ohne grosse Investition ein Abrechnungssystem einzuführen, welches einen bargeldlosen Bezug ermöglicht und zuverlässig den individuellen Konsum der Benutzer erfasst. Als Technologie wurde NFC festgelegt. Fast jede Person besitzt heutzutage bewusst oder unbewusst eine NFC-Karte. Sei es in Form eines Schlüssels, Mensakarte oder z.B. Gesundheitskarte. Es soll ein System erforscht werden, welches dem Benutzer ermöglicht mit einer oder mehreren NFC-Karten sich am System zu authentifizieren und Kaffee zu beziehen oder Wartungen an der Maschine durchzuführen.

Weiter wäre es wünschenswert wenn der letzte Konsument über Probleme an der Maschine automatisch und elektronisch informiert würden, bevor der nächste Konsument vor der nicht einsatzfähigen Maschine steht.

Auf den unterschiedlichen NFC-Karten sollten favorisierte Kaffeerezepte gespeichert werden können, sodass diese Rezepte automatisch bei der Authentifizierung übermittelt werden und die Kaffeemaschine direkt diesen gewünschten Kaffee herauslässt.

1.2 Ziel

Ziel dieses Forschungsprojektes ist es ein innovatives System zu erforschen, welches den Konsument authentifiziert und dessen Konsum speichert, den Status der Kaffeemaschine jederzeit überwacht und Servicemeldungen an die betroffenen Personen

weiterleitet, zudem sollten Kaffeerezepte automatisch und direkt an die Kaffeemaschine übermittelt und herausgelassen werden.

Im Rahmen dieser Arbeit wurden verschiedene Lösungsvorschläge erarbeitet und verglichen. Aus diesen Erkenntnissen wurde ein Lösungsvorschlag als Vorlage für den Prototypen ausgewählt. Neben den verschiedenen Lösungsvorschläge wurde die Interaktion und Bedienung des Systems genauer erforscht.

Zusätzlich zu den theoretischen erarbeiten Möglichkeiten wird im Rahmen dieses Projektes ein Prototyp entwickelt, welcher eine bargeldlose Bezahlung ermöglicht, den Status der Kaffeemaschine überwacht und eine Webserver, welcher Informationen wie konsumierten Kaffee und Servicearbeiten benutzerspezifisch visualisiert.

2 Grundlagen und Technologien

In diesem Abschnitt werden die Technologien vorgestellt, mit welchen ein Haushaltsgerät und im speziellen ein Kaffeefullautomat bedient und verwaltet werden kann. Neben den Grundlagen zu NFC wird das MDB-Protokoll erläutert, welches von vielen Haushalts- und Verkaufsgeräten für die Bezahlung und den eigentlichen Bezug von Produkten benutzt wird. Da dieses Protokoll beinahe unbekannt ist und neben der Spezifikation kaum Literatur vorhanden ist, wird das Protokoll detaillierter beschrieben.

2.1 NFC

Near Field Communication (NFC) ist ein Übertragungsstandard welcher kontaktlosen Austausch von Nachrichten über kurze Distanzen realisiert. Der NFC Standard wurde 2002 von NXP Semiconductors und Sony eingeführt. NFC wird als Untergruppe der Technologie RFID (Radio Frequency Identification) eingestuft, indem es standardisiert nur auf Distanzen bis maximal 10 cm betrieben werden kann, wo im Gegensatz RFID-Systeme Distanzen bis zu mehreren Meter überwinden können [9]. Die Datenübertragung basiert auf dem Protokoll ISO/IEC 14443 [16]. NFC-Systeme basieren wie gewisse RFID-Systeme auf die Übertragungstechnologie der induktiven Kopplung mit einer Betriebsfrequenz von 13,56 MHz. NFC-Systeme nutzen das Prinzip des Transformators zur Energie- und Datenübertragung. Die genau Funktionsweise und die dazu nötigen technischen Verfahren sind im „RFID-Handbuch“ [9] verständlich beschrieben. Neu bei NFC ist auch die strikte Trennung von Reader und Target. Klassische RFID-Systemen beinhalten immer eine aktive (Reader) und mehrere passive Komponenten (Target/Transponder), wo hingegen ein NFC-Gerät sowohl als aktive und passive Komponente (Reader und Target) betrieben werden kann. So können NFC Komponenten in unterschiedlichen Modi betrieben werden, neben dem typischen passiven Modus (Read-Write und Card-Emulation), in welchem der Reader ein Trägersignal zur Energie- und Datenübertragung bereitstellt, existiert ein aktiver Modus, in welchem Reader und Transponder (auch Reader) abwechselnd das Trägersignal zur Datenübertragung erzeugen (Peer-to-Peer Modus). In der Abbildung 1 sind die unterschiedlichen Betriebsmodi welche vom NFC-Forum definiert wurden abgebildet.

Im Rahmen dieses Forschungsprojektes wurden die Betriebsmodi Read-Write und Peer-to-Peer genauer analysiert und für die Entwicklung des Prototyps implementiert.

2.1.1 Read-Write Modus

Der Read-Write Modus ermöglicht die Kommunikation mit passiven RFID- und NFC-Transponder. Es wird die klassische Bitübertragungsschicht und Antikollisionsverfahren von RFID benutzt. Somit sind NFC-Systeme im Read-Write Modus mit RFID-Transponder kompatibel. Offiziell werden jedoch nur RFID-Tags, welche in Tabelle 1 aufgelistet sind vom NFC-Standard unterstützt.

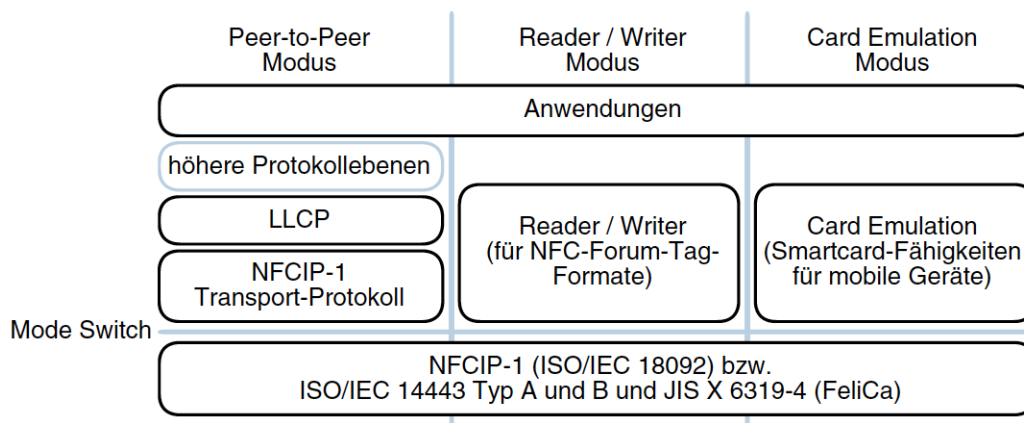


Abbildung 1: NFC Forum Architektur und Betriebsmodi [14]

Tag Format	RFID-Technologie	Hersteller	Hersteller	Speichergröße
Type 1	Topaz	Innovision	ISO 14443A	$\geq 120\text{Byte}$ $\leq 2048\text{Byte}$
Type 2	MIFARE Ultralight	NXP	ISO 14443A	$\geq 64\text{Byte}$ $\leq 2048\text{Byte}$
Type 3	FeliCa	Sony	JIS X 6319-4	$\geq 0\text{Byte}$ $\leq 1\text{MB}$
Type 4	Smartcards	-	ISO 14443A, ISO 7816-4	$\geq 0\text{Byte}$ $\leq 512\text{MB}$

Tabelle 1: NFC Forum unterstützte RFID-Transponder

Auf den in Tabelle 1 aufgelisteten Transponder können die gewöhnlichen Read-Write Operationen ausgeführt werden. Eine wichtige Funktion im Read-Write Modus ist das Auslesen der ID des Transponders. Anhand dieser ID wird ein Benutzer an der Kaffeemaschine authentifiziert. Die Lese- und Schreibfunktion vom Read-Write Modus wird zum Speichern der Kaffeerezepte benutzt. Es ist jedoch zu beachten, dass nicht auf allen Transponder ein Kaffeerezept gespeichert werden kann, da einige Transponder (Type 3 und 4) einen Lese- und Schreibschutz besitzen. Falls dieser Zugangsschlüssel unbekannt ist, können keine Daten auf den Transponder geschrieben oder von diesem gelesen werden. Die ID jedoch, kann von jeder Karte gelesen und zur Authentifizierung benutzt werden.

2.1.2 Peer-to-Peer Modus

Der Peer-to-Peer Modus ermöglicht den Datenaustausch zwischen zwei aktiven NFC-Geräten. Der Modus definiert einen Protokollstapel (Abbildung 2), welcher von beiden Geräten implementiert werden muss und vom NFC-Forum definiert wur-

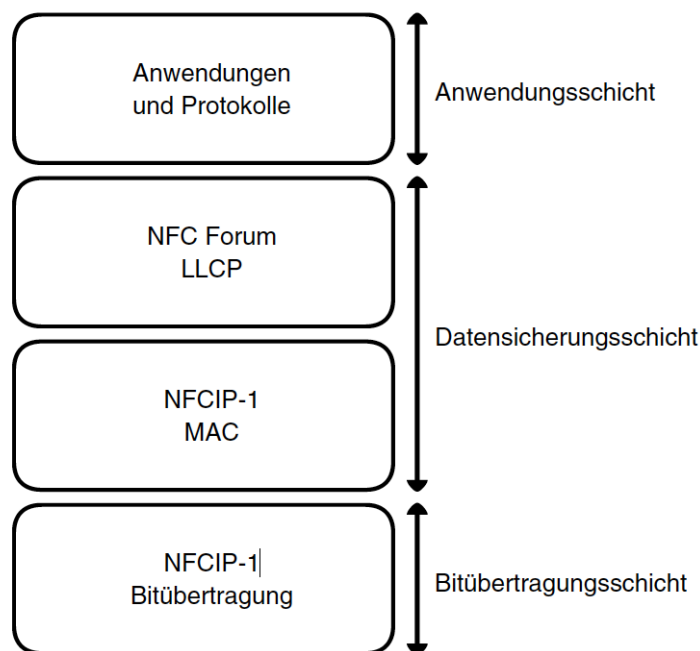


Abbildung 2: NFC Peer-to-Peer Modus Protokollstapel [14]

de. Die Bitübertragungsschicht ist im Standard ISO/IEC 18092 [1] definiert und bestimmt die Frequenz, Modulationen, Kodierung, und die Übertragungsgeschwindigkeit. Die Datenschichtung ist in den Media-Access-Control-Layer (MAC) und einen Logical-Link-Control-Layer (LLCP) unterteilt. Der MAC-Layer definiert das Verfahren für den Verbindungsaufbau und die Initialisierung des Datenaustauschprotokolls. Der LLCP Layer ermöglicht die Datenübertragung zwischen zwei logischen Endpunkten. In der Anwendungssicht können beliebige Daten zwischen den zwei beteiligten Geräten ausgetauscht werden.

Der Peer-to-Peer Modus wird häufig bei NFC fähigen Mobiltelefonen eingesetzt. Im Rahmen dieses Forschungsprojektes wird der Peer-to-Peer Modus verwendet für den Austausch der Authentifizierungsdaten zwischen Mobiltelefon und dem Server, welcher den Bezug von Kaffee steuert.

2.2 MDB

Multi Drop Bus (MDB) ist ein Protokoll welches den Zahlungs- und Abrechnungsprozess von Verkaufsautomaten definiert. Die erste Version des Protokolls wurde 1993 eingeführt [7]. Das MDB-Protokoll hat sich in diesem Bereich der Verkaufsautomaten etabliert. So werden heute die meisten Verkaufsautomaten mittels dem MDB-Protokoll gesteuert. Neben den klassischen Verkaufsautomaten für Bahntickets, Getränke und Verpflegung, ist das MDB-Protokoll immer mehr an Haushaltsgeräten wie z.B. Kaffeevollautomaten anzutreffen. Das MDB-Protokoll hat einige spezielle Eigenheiten, welche es verunmöglicht ein MDB-Gerät direkt an einen

Computer anzuschliessen (Abschnitt 2.2.4).

2.2.1 Funktionsweise

Das MDB-Protokoll definiert einen seriellen Bus, welcher mit einer 9600 Bitfrequenz im Master-Slave Konzept betrieben wird. Auf dem MDB-Bus existiert genau ein Master, welcher auch Vending Mechanism Controller (VMC) genannt wird. Alle anderen Teilnehmer werden als Slaves betrieben. Auf dem Bus sind maximal 32 Slaves aktiv und adressierbar. Jedem Slave wird eine einmalige, voreingestellte Adresse zugewiesen, über welche er vom Master in regelmässigen Abständen „gepollt“ wird. Die Kommunikation zwischen Master und Slave basiert auf gezielte Anfragen des Master an die Slaves („Polling“). Die Slaves antworten ausschliesslich und in vordefinierten Zeiträumen auf die „Polls“ des Masters. Wenn ein Slave nicht in dem vorgegebenen Zeitfenster antwortet, wird angenommen, dass der Slave nicht angeschlossen oder fehlerhaft ist und wird deshalb vom Master deaktiviert. Die Slaves können sich in unterschiedlichen Zustände (Abbildung 3) befinden. Der Master kennt die Zustände der Slaves und schickt ihnen die zustandsspezifischen Kommandos zu. Übertragungsfehler werden anhand einer Prüfsumme, welche jeder Nachricht hinzugefügt wird, erkannt und können falls nötig mit Hilfe eines Wiederholungskommandos neu gesendet werden. [6]

Es existieren mittlerweile sehr unterschiedliche MDB-Devices (Slaves). Am häufigsten sind nach wie vor die klassischen Münz- und Notenautomaten anzutreffen, es existieren jedoch mittlerweile auch Cashless Devices (ab MDB Version 3), welche mit Techniken wie RFID/NFC arbeiten können (Abschnitt 3.2). Die MDB-Devices können in verschiedenen Betriebslevel (1 bis 3) betrieben werden. Diese Level definieren den Funktionsumfang des angeschlossenen MDB-Devices. Damit die MDB-Systeme rückwärts kompatible sind müssen auf dem MDB-Master alle Betriebslevel unterstützt werden. Die MDB-Devices teilen der VMC beim Setup-Prozess den unterstützten MDB-Level mit. In den höheren Levels können komplexere Befehle wie Displayanzeigen (Level 2) oder Dateiaustausch (Level 3) realisiert werden.

2.2.2 Protokoll Eigenschaften

Wie erwähnt hat das MDB-Protokoll sehr spezifische Eigenschaften, welche stark von etablierten Standards abweichen. So werden Daten im MBD-Protokoll in einem 9 Bit Paket übermittelt. (Tabelle 2)

	8 Data Bits									
	1 Mode Bit									

	9 Bits Total									
LSB										MSB
Start	0	1	2	3	4	5	6	7	Mode	Stop

Tabelle 2: Serial Bit Format MDB

Das „Mode Bit“ definiert den Inhalt der „Data Bits“. Ist das „Mode Bit“ gesetzt, befindet sich in den „Data Bits“ eine Adresse, welche von allen Slaves gelesen werden muss. Ist das „Mode Bit“ nicht gesetzt, beinhaltet die „Data Bits“ MDB-Kommandos und werden nur vom adressierten Slave gelesen.

Die Kommandos zwischen Master und Slave werden Blockweise übermittelt. Ein Block besteht aus einem Address-Byte, gefolgt von optionalen Data-Bytes und einer Prüfsumme. Der Master wie auch der Slave bestätigen das Erhalten des Blocks mittels einem ACK oder einem NAK falls der Block nicht erhalten wurde. Mittels RET kann der Master ein Block vom Slave erneut anfordern.

Weiter ist zu beachten, dass ein Slave dem Master immer innerhalb des Sendezeitraums von 5 ms antworten muss. Kriegt der Master während dieses Sendezeitraums keine Antwort vom Slave wird der Slave vom Master aufgefordert den Block erneut zu senden (RET). Falls ein Slave innerhalb von 100ms nicht antwortet wird der Slave zurückgesetzt und in den Status „Disabled“ versetzt. Der Master „pollt“ die Slaves alle 20-200 ms.

Der Master versorgt die Slaves mit Strom, deshalb benützen MDB-Slaves keine eigene Stromversorgung. Der Master versorgt die Slave mit einer maximalen Ausgangsspannung von 34V. Zudem definiert die MDB-Spezifikation spezielle Kabel und Stecker für die Hardware von MDB-Geräte, welche sich stark von den gängigen Stecker unterscheiden und somit nicht direkt an einen Computer angeschlossen werden können. [6]

2.2.3 Cashless Device

Im Rahmen dieses Forschungsprojektes werden die MDB Cashless Devices genauer analysiert. Falls RFID/NFC und MDB eingesetzt wird, werden die NFC-Geräte immer als Cashless Devices betrieben. Besonders gilt es zu beachten, dass Cashless Devices einem klar definierten Zustandsmodell entsprechen müssen. Die MDB-Kommandos welche zwischen Master und Slave übertragen werden, hängen stark von den Zuständen des Slaves ab. Die Cashless Device Zustände und die verschiedenen Kommandos, welche die Zustandsübergänge definieren sind in Abbildung 3 dargestellt.

Auf die MDB-Kommandos wird in dieser Dokumentation nicht genauer eingegangen, da es den Rahmen dieser Arbeit sprengen würde. Die Kommandos können jedoch in der Spezifikation [6] nachgelesen werden. Wichtig ist jedoch, dass die Kommandos und Zustände genau wie in der Spezifikation vorgegeben implementiert werden, da sonst keine Kommunikation zwischen Master und Slave zustande kommen kann.

Das Cashless Device ist nur in gewissen Zuständen fähig eine Bezahlung durchzuführen. In der Tabelle 3 sind die verschiedenen Zustände und ihre Möglichkeiten bezüglich des Konsums abgebildet. Betriebsbereit bedeutet, dass der Slave vom Master für Zahlungen akzeptiert wurde. Bezugsbereit bedeutet, dass ein Benutzer erfolgreich authentifiziert wurde und eine Verkauf stattfinden kann.

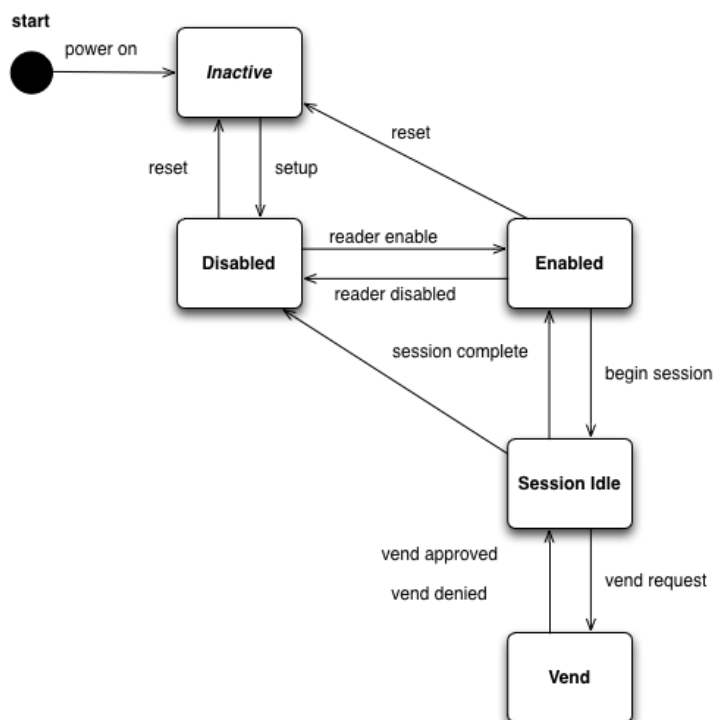


Abbildung 3: MDB Cashless Device Zustandsdiagramm

Status	Betriebsbereit	Bezugsbereit
Inactive	Nein	Nein
Disabled	Nein	Nein
Enabled	Ja	Nein
Session Idle	Ja	Ja
Vend	Ja	Ja

Tabelle 3: MDB Cashless Device Betriebs- und Bezugsbereit Übersicht

2.2.4 MDB2PC

Wegen den speziellen Eigenschaften (Abschnitt 2.2.2) des MDB-Protokolls kann ein Computer nicht ohne weiteres an einen MDB-Bus angeschlossen werden. Die Zeitfenster, die spezielle Kommandostruktur, die Stromversorgung und nicht zuletzt die unüblichen Stecker erfordern spezielle Hardware. Diese Hardware muss selbst gebaut oder von einem Hersteller, welcher solche MDB2PC Hardware produziert, gekauft werden. Für die Realisierung des Prototypen wurde ein MDB2PC Interface der Firma Abrantix AG erworben. Dieses Interface übernimmt die Kommunikation mit dem MDB-Master und sendet die MDB-Kommandos über die serielle Schnittstelle weiter an einen Computer (Abbildung 4). Das MDB2PC Interface antwortet dem Master innerhalb der Zeitfenster und speichert die MBD-Kommandos lokal zwischen, welche



Abbildung 4: MDB2PC Interface Protokoll [8]

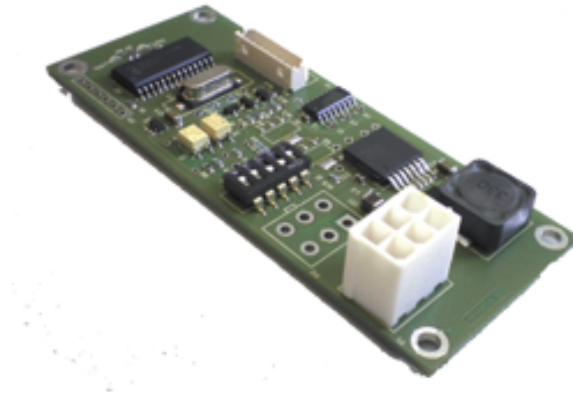


Abbildung 5: MDB2PC Interface [2]

dann leicht verzögert an den Computer weiter geleitet werden.

Das MDB2PC Interface regelt die MDB spezifischen Protokolleigenschaften. Das Zustandsdiagramm sowie das Lesen und Versenden der MDB-Kommandos muss jedoch vollständig implementiert werden (Abschnitt 7.5).

3 Existierende Systeme für die Abrechnung und Verwaltung von Kaffeemaschinen

Es existieren bereits viele innovative Systeme für Kaffeemaschinen. Dies hat sicher mit dem Klischee des Kaffee trinkenden Informatikers zu tun. So wurde z.B. der Vorläufer der „Web-Kamera“ 1991 von Quentin Stafford-Fraser erfunden. Er richtete eine Kamera auf die Kaffeekanne und übertrug das Bild auf einen Server. So konnte er erkennen ob noch Kaffee verfügbar war, um nicht vergebens an die Kaffeekanne zu gehen [18]. In den Recherchen bezüglich Bedienung und Verwaltung von Kaffeemaschinen ist auffällig, wie viele Systeme die einzelnen Aufgaben: Abrechnung, Überwachung und automatischer Bezug von Kaffee realisieren. Für die Abrechnung ist eine grosse Auswahl an Produkten erhältlich. Diese sind in unterschiedlicher Form zu erwerben. Für die Verwaltung von Kaffeemaschinen und die Konfiguration von Kaffeerezepten gibt es vereinzelt herstelleraufspezifische Lösungen. Zudem gibt es ein Projekt der Firma Zipwhip, welches eine automatische Konfiguration und den Bezug von Kaffee realisiert. Das einzige System welches alle Funktionalitäten in Kombination mit RFID in einem System beherbergt, ist die „Smart Saeco“ von der TU Darmstadt.

3.1 Externe Steuerung und Verwaltung

In meinen Recherchen habe ich einige Systeme gefunden, welche die Kaffeemaschine extern steuern und überwachen lässt. Einzelne Anbieter stellen solche Hard- und Software zur Verfügung.

3.1.1 Jura Web Pilot

Der Kaffeemaschinen Hersteller Jura Gastro World hat zu dem Kaffeemaschinen Modell Impressa 500 ein Web-Interface (Jura Web Pilot) entwickelt. Über dieses Web-Interface kann man den Status der Kaffeemaschine jederzeit über das Internet abfragen. Neben der Statusanzeige ermöglicht einem der Web Pilot die Tasten an der Kaffeemaschine mit Rezepten zu belegen. So kann der Administrator Kaffeerezepte konfigurieren, jedoch werden die Rezepte und Tastenbelegungen nicht Benutzer spezifisch konfiguriert sondern sind bei Änderungen für alle Benutzer gültig. Leider ist der Jura Web Pilot nur mit einem einzigen Modell von Jura kompatibel und wurde nicht auf andere Modelle portiert.

Mit Hilfe des Web Pilot können auch Statistiken der Kaffeemaschine abgerufen werden, wie z.B. Anzahl bezogene Kaffees, Anzahl Entkalkungen, usw. So können die Techniker von Jura über das Internet Diagnosen und Softwareupdates an der Kaffeemaschine durchführen.

Das Web Interface wird am Service-Port der Kaffeemaschine angeschlossen. Der Web Pilot überträgt die über den Service-Port empfangenen Daten wireless an einen Computer weiter, welcher wiederum mit dem Internet verbunden ist und die Daten der Kaffeemaschine ins Internet stellt. Die gesamte Kommunikation ist verschlüsselt,

sodass ein abfangen und analysieren der Daten nicht möglich ist. [11]

3.1.2 Textspresso von Zipwhip

Zipwhip ist ein SMS Service-Anbieter für Mobiltelefone. Zu Werbezwecken haben die Ingenieure ein SMS gesteuertes Kaffeebezugsystem entwickelt. Dieses ermöglicht es den Mitarbeiter, eine SMS mit dem gewünschtem Kaffee (Espresso, Americano, usw.) an die Kaffeemaschine zu senden. Die Kaffeemaschine lässt dann automatisch diesen per SMS gesendeten Kaffee heraus. Zusätzlich wird mit einem manipulierten Drucker die Telefonnummer mit Schokoladenpulver auf den Kaffee gedruckt. Damit dieses System realisiert werden konnte, mussten diverse Komponenten entwickelt werden. Das System besteht aus einer zentralen Logikeinheit, welche die SMS empfängt und diese analysiert. Dann wird ein Arduino-Roboter aufgefordert, eine Tasse von einem Tablett zu nehmen und unter die Kaffeemaschine zu schieben. Ist die Tasse unter der Kaffeemaschine positioniert, löst die Logikeinheit ein Knopfdruck aus. Dafür wurden die Knöpfe der Kaffeemaschine entfernt und mit Kabel über ein Relaischaltung mit der Logikeinheit verbunden. Ist der Kaffee bezogen bewegt der Roboter die Tasse weiter unter einen modifizierten Tintenstrahldrucker, welcher anstelle von Tinte Schokoladenpulver druckt. So wird die Absendernummer mit Schokoladenpulver auf den Kaffee gedruckt. Danach wird der Kaffee vom Roboter auf eine Ablagestation gestellt, auf welcher der Kaffee vom Absender abgeholt werden kann. [4]

Dieses Kaffeesystem demonstriert Ansätze, welche im Rahmen dieses Forschungsprojektes genutzt werden können. Sehr interessant ist der automatische Bezug des Kaffees. Zudem kann der Bezug durch die Identifikation der Telefonnummer des Absenders abgerechnet werden. Eine Statusüberwachung ist mit diesem System jedoch nicht möglich, da die Nachrichten auf dem Display der Kaffeemaschine nicht analysiert werden.

3.2 Abrechnungssysteme

Die meisten Abrechnungssysteme für Kaffeevollautomaten werden mittels dem MDB-Protokoll (Abschnitt 2.2) realisiert. Fast alle professionellen Kaffeevollautomaten besitzen ein MDB-Interface oder die Möglichkeit ein solches anzuschliessen. Auch neuere Technologien wie NFC verwenden für die Kommunikation und die eigentliche Abrechnung und den Bezug der Kaffees das MDB-Protokoll. Neben den NFC basierten Systemen (bargeldlose Systeme) werden sehr häufig Münz- und Notenautomaten eingesetzt.

3.2.1 Jura MDB-Interface

Da Jura auch eine grosse Palette an professionellen Kaffeevollautomaten herstellt, bietet Jura ein MDB-Interface für ihre Kaffeemaschinen an. Dieses MDB-Interface ermöglicht Kunden ihre Kaffeemaschine über das MDB-Protokoll zu steuern. Dies bedeutet, dass ein üblicher Bezug von Kaffee nicht mehr möglich ist. Wählt der



Abbildung 6: Textpresso von Zipwhip [3]

Kunde ein Kaffee aus, wird auf dem Display der Kaffeemaschine der nötige Betrag für den Bezug des Kaffees angezeigt. Der Kunde muss nun über ein proprietäres System (Münz- Notensystem, bargeldloses System) diesen Betrag bezahlen. Das externe System teilt dann der Kaffeemaschine über das MDB-Protokoll mit, ob der Benutzer genügend Geld eingeworfen hat. Das Jura MDB-Interface wird als MDB-Master betrieben. Das Jura MDB-Interface bietet somit die Möglichkeit den Kaffeebezug mit einer Abrechnungssoftware zu verbinden. Leider werden über das MDB-Interface keine Statusnachrichten übermittelt. So kann z.B. über das MDB-Protokoll nicht erkannt werden, ob die Kaffeemaschine kein Wasser oder Kaffee mehr hat. Das Jura MDB-Interface arbeitet laut Spezifikation im MDB-Level 2. Jedoch sind viele dieser Funktionalitäten im Jura MDB-Interface nicht vollständig implementiert, sodass Displayanfragen des MDB-Slaves nicht auf dem Display der Jura Kaffeemaschine dargestellt werden.

Weiter kann man über eine Software, welche von Jura zur Verfügung gestellt wird, den MDB-Modus ein- und ausschalten. Mittels der Software kann man die Preise der verschiedenen Kaffees festlegen und den Display der Kaffeemaschine anpassen. Die Software kommuniziert über die serielle Schnittstelle mit dem MDB-Interface. Da die Verbindung nicht verschlüsselt ist, können die Kommandos abgefangen werden und in eigenen Anwendungen benutzt werden, um das Jura MDB-Interface zu konfigurieren (z.B. MDB-Modus ein- und ausschalten). [5]

3.2.2 Münz- und Notensystem

Im MDB-Umfeld werden am häufigsten Münz- und Notensysteme als Zahlungssystem eingesetzt. Diese Art von Bezahlung ist eine der ältesten und die wohl am weitverbreitetsten Variante. Deshalb sind auch sehr unterschiedliche Automaten auf dem Markt erhältlich. Beim Kauf eines Münz- Notensystems ist darauf zu ach-

ten, dass diese währungsabhängig sind und nur selten unterschiedliche Währungen unterstützen. Da Münz- und Notenautomaten oft im öffentlichen Raum aufgestellt werden, sollte ein besonderes Augenmerk auf Diebstahlschutz und Vandalismus gesetzt werden.

Die Münz- Notenautomaten werden als MDB-Slave im MDB-Level 1 betrieben. Somit muss keine grosse Logik in den Automaten realisiert werden. Deshalb sind Münzautomaten schon für einen erschwinglichen Betrag käuflich. Es existieren auch Münzautomaten, welche nicht reales Geld entgegen nehmen, sondern mit eigenen Jettons arbeiten. Diese Art von Münzautomaten, wäre in unserem Anwendungsfall denkbar. Jedoch sehr unpraktikabel, da jemand diese Jettons verwalten und herausgeben müsste. Zudem sind diese Jettons nicht personalisiert und müssten bei Herausgabe im System manuell erfasst werden.

3.2.3 Bargeldlose Systeme

Eine weitaus praktikablere Variante der Bezahlung mittels MDB sind die bargeldlosen Systeme. Auf dem Markt sind unterschiedliche bargeldlose Systeme vorhanden. Häufig werden diese Systeme mit eigenen RFID-Transponder ausgeliefert, auf welchen ein gewisser Geldbetrag, welcher auch an einem Münz- oder Notenautomaten aufgeladen werden muss, gespeichert wird. Diese Systeme sind jedoch teuer in der Beschaffung und somit nicht geeignet für kleinere Unternehmen.

Durch die Integration von NFC-Reader in Mobiltelefonen werden bargeldlose Abrechnungssysteme immer beliebter. Diverse Banken und Mobilfunkbetreiber bieten ihren Kunden die Möglichkeit mittels NFC zu bezahlen. Eine Anbindung an diese Dienstleister, wäre eine Möglichkeit die bargeldlose Bezahlung zu realisieren. Damit man jedoch Geld der Kunden bei den Dienstleister beziehen kann, muss man sich meist als vertrauenswürdiger Partner registrieren. Dies ist leider noch nicht ohne grösseren Aufwand und Kosten möglich. [17]

In meinen Recherchen habe ich kein bargeldloses System entdeckt, welches eine Bezug von Produkten mittels der ID diversen NFC-Karten und ohne den Inhalt der Karten zu beschreiben oder zu lesen funktioniert. Sodass Benutzer ihre vorhandenen NFC-Karten für den Bezug der Produkte nutzen können.

3.3 Smart Saeco

Smart Saeco ist das einzige System welches alle Anforderungen (Abrechnung, Verwaltung und automatischer Bezug) in einem System realisiert. Das System wurde von Entwicklern der Technischen Universität Darmstadt im Rahmen der Forschung im Bereich „smart living“ umgesetzt. Die Motivation ein solches System zu entwerfen, war die gleiche wie jene dieses Forschungsprojektes.

Die Benutzer können sich mittels ihrem Türschlüssel an der Kaffeemaschine authentifizieren. Der persönliche Kaffeekonsum wird in einer Datenbank gespeichert. Über eine Homepage kann der persönliche Konsum überprüft werden. Zudem wurde die Kaffeemaschine so manipuliert, dass ein Bezug des gewünschten Kaffees automatisch bezogen wird. Die Kaffeetassen wurden mit RFID-Tags versehen und un-

3 EXISTIERENDE SYSTEME FÜR DIE ABRECHNUNG UND VERWALTUNG VON KAFFEEMASCHINEN

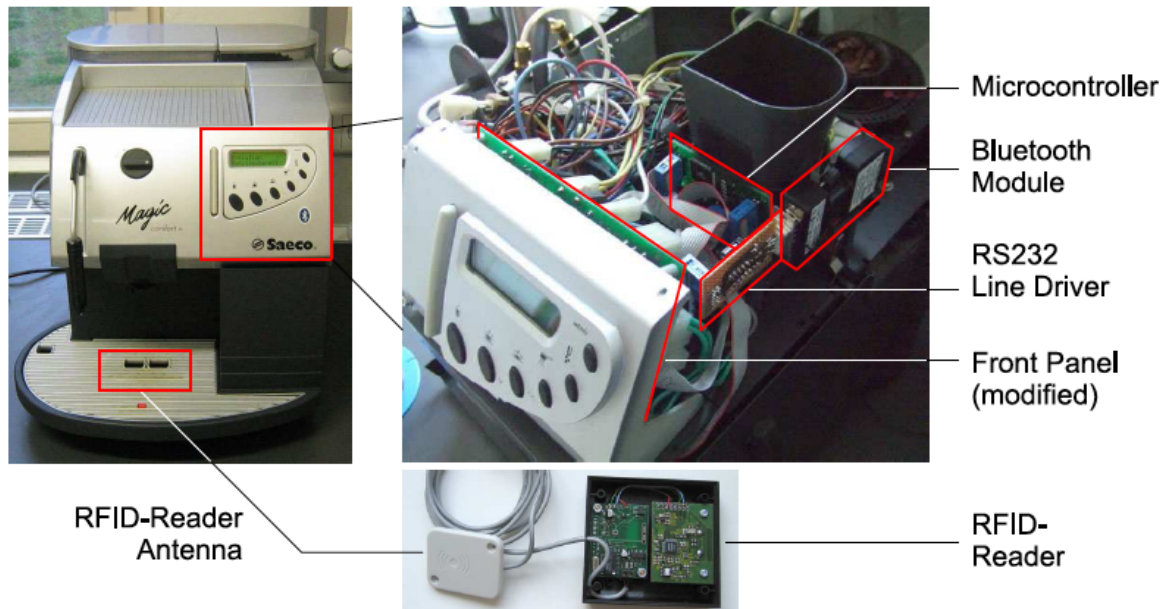


Abbildung 7: Smart Saeco Hardware Modifikationen [10]

ter der Kaffeemaschine wurde ein RFID-Reader installiert. Der Reader erkennt die Kaffeetasse und kann den automatischen Bezug auslösen. Weiter werden Statusmeldungen der Kaffeemaschine über Text-to-Speech dem Benutzer mitgeteilt. Falls eine Reinigung durchgeführt werden muss, wird der Benutzer von der Kaffeemaschine angeleitet diese Wartungsarbeiten durchzuführen. [15]

Die gesamte Funktionalität des Systems basiert auf der Manipulation und Änderung der Hardware von der Kaffeemaschine (Abbildung 7). Dadurch ist dieses System leider nur an dieser Maschine verwendbar und nicht adaptierbar auf andere Haushaltsgeräte oder Kaffeemaschinen. Jedoch werden die unterschiedlichen Funktionalitäten sehr schön demonstriert.

4 System Anforderungen

Damit das gewünschte System entworfen werden kann, müssen diverse funktionale und nichtfunktionale Anforderungen definiert sein. In diesem Kapitel werden die gewünschten Anforderungen aufgelistet und die Möglichkeiten analysiert, wie diese zu realisieren sind. In den folgenden Kapiteln werden die verschiedenen Lösungsvorschläge, unter spezieller Betrachtung der festgelegten Anforderungen erforscht.

4.1 Nichtfunktionale Anforderungen

In diesem Abschnitt werden wichtige nichtfunktionale Anforderungen genauer beschrieben. Neben den hier beschriebenen nichtfunktionalen Anforderungen sollten alle anderen in der Realisierung auch berücksichtigt und erfüllt werden, wie:

- Sicherheitsanforderung
- Zuverlässigkeit
- Korrektheit
- Aussehen und Handhabung
- Leistung und Effizienz
- Skalierbarkeit
- usw.

4.1.1 Flexibilität

Eine wichtige nichtfunktionale Anforderung ist die Flexibilität. Dies bedeutet, dass übliche Standards unterstützt werden und das System in verschiedenen Umgebungen funktionsfähig bleibt. Das zu entwerfende System sollte so gestaltet werden, dass es adaptierbar auf andere Haushaltsgeräte sowie unterschiedliche Modelle und Hersteller von Kaffeemaschinen ist.

Das MDB-Protokoll ist sehr flexibel, da es ein weit verbreiteter Standard für Abrechnungs- und Bezahlssysteme ist. Die meisten Kaffeemaschinenhersteller unterstützen das MDB-Protokoll.

Der Ansatz, welcher im Projekt Smart Saeco (Abschnitt 3.3) gewählt wurde, eine hersteller- und modellspezifische Lösung zu entwickeln ist nicht flexibel. In unserer Umgebung arbeiten wir mit einer Jura XF 50. Auch hier wurde die Überlegung eine modell- und herstellerspezifische Lösung zu entwickeln diskutiert. Diese Ansatz wurde jedoch wieder schnell verworfen, da das Protokoll der Jura Kaffeemaschinen nicht öffentlich zugänglich ist. Die Anfrage auf die Dokumentation des Protokolls für diese Arbeit wurde abgelehnt. Somit wäre eine solche Lösung nur möglich, wenn das Protokoll entschlüsselt würde. Diese hätte jedoch zur Folge dass die Arbeit aus rechtlichen Gründen nicht veröffentlicht werden könnte.

4.1.2 Kosten

Eine weitere wichtige nichtfunktionale Anforderung sind die Randbedingungen (Kosten), diese sollten so gering wie möglich gehalten werden. Das Ziel dieses Forschungs-

projektes ist es ein System zu entwerfen, welches von kleineren Unternehmen ohne grosse Investitionen angeschafft werden kann.

Die Software sollte daher komplett Open Source sein, damit keine Lizenzgebühren anfallen und die Software erweitert und verändert werden kann.

Die Hardwarekosten sind stark abhängig von dem Gerätehersteller, denn der teuerste Kostenpunkt ist die MDB-Hardware. Zwingend notwendig ist ein MDB2PC Interface, welches von unterschiedlichen Hersteller gekauft werden kann. Die Baupläne für ein solches MDB2PC Interface sind auch in der Spezifikation abgebildet, sodass ein solches Interface auch selbst gebaut werden kann.

4.1.3 Benutzbarkeit

Die Benutzbarkeit des System wurde als wichtige nichtfunktionale Anforderung identifiziert. Speziell bei NFC, welches eine nicht visuelle Kommunikation realisiert ist die Interaktion mit dem Benutzer sehr wichtig, da nicht ersichtlich ist ob ein Prozess stattgefunden hat oder nicht. Der Aufwand das System zu bedienen sollte so gering wie Möglich sein, da sonst die Benutzer frustriert sind und die Maschine nicht benutzen. Durch den Mehraufwand der Registrierung und jeweiliger Authentifizierung sollte dem Benutzer eine Gegenleistung erbracht werde, damit er den Mehraufwand gerne in Kauf nimmt.

Die Möglichkeiten der Bedienung des System wurden detaillierter in Abschnitt 6 betrachtet.

4.2 Funktionale Anforderungen

Die funktionalen Anforderungen definieren die eigentliche Funktionalität des Systems. Wie einleitend schon erwähnt wurde, ist ein Komplettsystem zu entwerfen, welches Abrechnung, Verwaltung und automatische Konfiguration und den Bezug von Kaffee ermöglichen sollte. In den folgenden Unterkapiteln werden die verschiedenen Funktionalitäten beschrieben.

4.2.1 Abrechnung

Es sollte automatisch und zuverlässig der Konsum der verschiedenen Benutzer erfasst und gespeichert werden. So kann bei Bedarf der individuelle Konsum überprüft werden. Es sollten Statistiken erstellt werden, auf welchen als Auslöser ein Benutzer aufgefordert werden kann, Kaffeebohnen zu besorgen oder Servicearbeiten an der Kaffeemaschine vorzunehmen.

Die Maschine sollte normalerweise gesperrt sein, damit ein unautorisierter Bezug von Produkten nicht möglich ist. Authentifiziert sich ein Benutzer kann er ein Produkt konsumieren, dieser Konsum wird in einer Datenbank gespeichert. Sobald der Benutzer seine NFC-Karte vom Reader nimmt sollte die Maschine wieder gesperrt sein und keinen Konsum, bis zur nächsten Authentifizierung zulassen.

Zur Authentifizierung sind verschiedene Techniken denkbar. So könnte man durch einen übliche „Username-Passwort“ Abfrage die Kaffeemaschine entsperren. Dies ist

jedoch aufwändig, da der Benutzer jedes Mal sein Passwort eingeben müsste. Dieser Vorgang ist zu zeitintensiv und somit nicht praktikabel.

Eine weitere Möglichkeit wäre die Authentifizierung mittels biometrischen Daten. Dieser Ansatz ist durchaus interessant, da durch z.B. Finger- oder Iriserkennung eine schnelle Authentifizierung möglich wäre. Jedoch ist hier der Datenschutz zu beachten, da vermutlich nur wenige Benutzer bereit sind ihrer sensiblen, persönlichen biometrischen Daten in einer Datenbank, welche für den Kaffeekonsum benutzt wird zu hinterlegen.

NFC als die gewählte Technik, ist hierfür am geeignetsten. Der Benutzer muss nur seine Karte auf einen Reader halten und kann dann bei erfolgreicher Authentifizierung Kaffee konsumieren. Es bietet sich auch an mehrere NFC-Karten an einen Benutzer zu binden. Somit können verschiedene Kaffeerezepte auf verschiedenen NFC-Karten hinterlegt werden. Da heutzutage fast jeder bewusst oder unbewusst eine NFC-Karte besitzt, müssen keine neue NFC-Karten herausgegeben werden. Falls jemand doch keine NFC-Karte besitzt, können diese kostengünstigen Karten vom Administrator bezogen werden.

Als Technik für das Sperren und Entsperren der Kaffeemaschine eignet sich das MDB-Protokoll gut, da dieses Protokoll weit verbreitet und somit flexibel ist. Zudem werden die Anforderungen für das Sperren und Entsperren der Maschine erfüllt.

4.2.2 Verwaltung

Unter Verwaltung der Kaffeemaschine ist die Administration der Maschine gemeint. Dies bedeutet, dass Probleme an der Kaffeemaschine erkannt werden und an die beteiligten Personen weitergeleitet werden, damit die Verursacher der Probleme diese beheben können. Zudem sollten Administratoren in der Lage sein, das gesamte System zu aktivieren und zu deaktivieren.

Damit die Kaffeemaschine verwaltet werden kann, muss der Status der Kaffeemaschine überwacht werden. Die meisten Kaffeemaschinen informieren die Benutzer über einen eingebauten Display. Einige Systeme wie der Jura Web Pilot ermöglicht eine Status-Benachrichtigung über das Internet. Dies ist jedoch wieder eine herstellerspezifische Lösung und dadurch nicht geeignet für unsere Systementwurf. Ein weiterer Ansatz wäre, die Signale welche von der Maschine an den Display gesendet werden abzufangen, zu analysieren um so den Status der Kaffeemaschine zu erfassen. Damit diese Signale analysiert werden können, wäre ein gutes Oszilloskop notwendig.

Eine weitere Möglichkeit ist das Fotografieren des Displays. So kann mittels Texterkennung der Zustand der Kaffeemaschine erfasst werden. Diese Variante hat den Vorteil, dass sie sehr flexibel ist und bei allen Kaffeemaschinen oder andern Haushaltsgeräten funktionieren wird. Die Transformationen und Filter auf die Bilder für die Texterkennung müssen je nach Maschine und Lichtverhältnisse angepasst werden. Zudem ist diese Variante sehr günstig, da keine hochauflösende Kamera dafür benötigt wird.

Ist der Status der Kaffeemaschine erfasst, muss dieser in der Datenbank gespeichert werden. Zudem sollten Fehlermeldungen erkannt und dem Kaffeadministrator

gemeldet werden. Denkbar sind Szenarios wie z.B. dass der letzte Benutzer, welche den Fehler verursacht hat benachrichtigt wird, dass er doch bitte diesen Fehler beheben soll.

Weiter sollte bei der Authentifizierung erkannt werden ob ein Problem an der Kaffeemaschine besteht und dieses beim Verlassen der Maschine gelöst ist. Ist dies der Fall soll diesem Benutzer in der Datenbank seine Servicearbeit gutgeschrieben werden.

4.2.3 Automatische Konfiguration und Bezug von Kaffee

Eine weiter wichtige funktionale Anforderung ist die Konfiguration und der automatische Bezug von favorisierten Kaffeerezepten. Der Benutzer sollte die Möglichkeit haben seine gewünschten Kaffeerezepte einer NFC-Karte zuzuweisen. So wird bei der Authentifizierung mit der NFC-Karte automatisch der bevorzugte Kaffee bezogen. Es muss dem Benutzer die Möglichkeit geboten werden jederzeit sein priorisiertes Rezept zu ändern. Zudem ist in einem weiteren Schritt geplant einen Mobiltelefon-Applikation zu schreiben, auf welcher eine virtuelle Kaffeemaschine abgebildet ist. So kann der Benutzer die Applikation auf dem Weg zur Kaffeemaschine starten und seinen gewünschten Kaffee vorkonfigurieren. Danach kann der Benutzer sein Mobiltelefon an den NFC-Reader halten und wird über den NFC Peer-to-Peer Modus authentifiziert und das vorkonfigurierte Rezept wird an die Kaffeemaschine übermittelt und automatisch bezogen.

Damit eine externe Konfiguration und der automatische Bezug des Kaffees ermöglicht werden kann, muss die Kaffeemaschine manipuliert werden. Sprich, man wählt wieder den Ansatz, dass das Protokoll des Herstellers bekannt ist. Dieser Ansatz ist dann jedoch wieder sehr inflexibel und nicht adaptierbar auf andere Systeme. Eine flexiblere Lösung ist das anlöten eines Kabels an die Knöpfe der Maschine. Diese können dann über eine Relaischaltung gesteuert werden und es kann so ein Knopfdruck simuliert werden. Dieser Ansatz wurde auch bei dem Projekt Textpresso (Abschnitt 3.1.2) realisiert. Eine dritte Variante ist die Montage eines Roboterarms vor der Kaffeemaschine. Der Roboterarm kann einfach für neue Maschinen kalibriert werden und ist dadurch sehr flexibel. Jedoch nimmt ein solcher Roboterarm sehr viel Platz vor oder neben der Kaffeemaschine ein und die Maschine wäre nur noch schwer von Hand bedienbar. Zudem bewegen sich die Roboterarme nur langsam, somit würde einen Bezug von Kaffee erheblich länger dauern als der direkte Bezug per Knopfdruck.

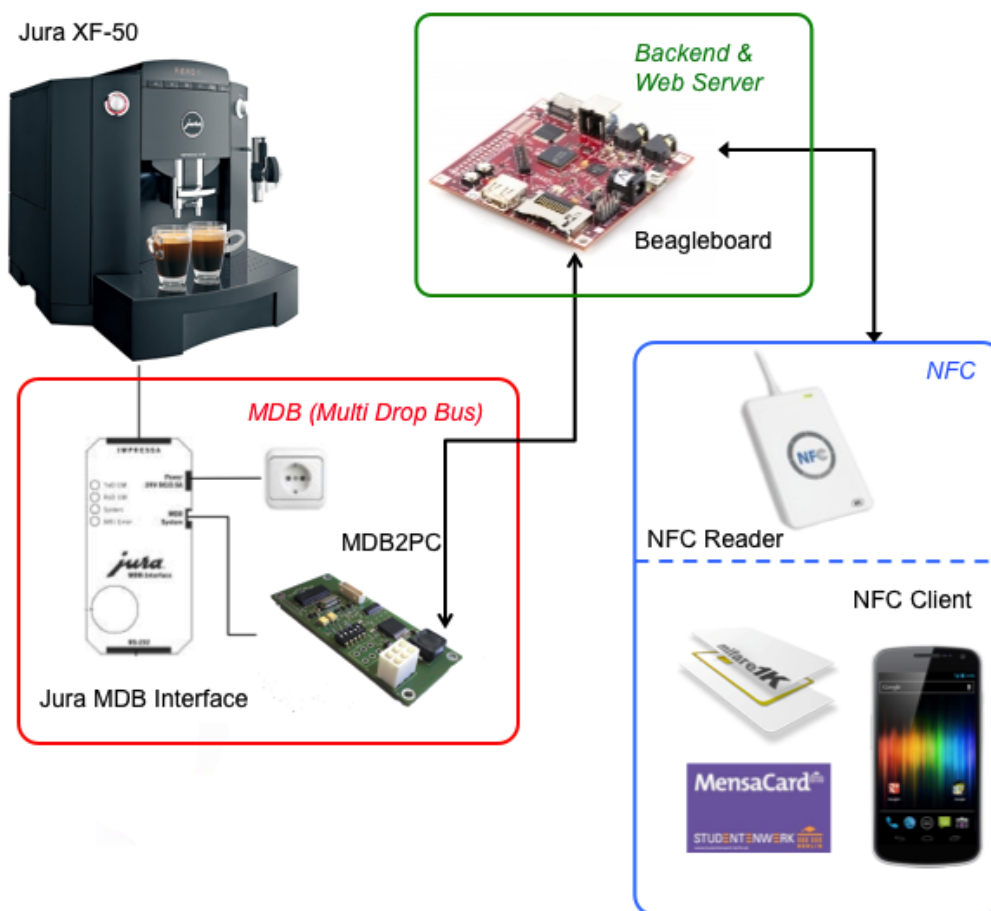


Abbildung 8: Lösungsvorschlag 1 - MDB

5 Lösungsvorschläge

In diesem Abschnitt werden die möglichen Lösungsvorschläge, unter besonderer Betrachtung der aufgestellten Anforderungen (Abschnitt 4), für einen möglichen Systementwurf analysiert. Es werden die Vor- und Nachteile der unterschiedlichen Vorschläge erarbeitet. Anhand der Erkenntnisse wird ein Lösungsvorschlag gewählt, welcher im Rahmen eines Prototypen in diesem Forschungsprojekts entwickelt wurde. Das direkte Ansprechen des Protokolls der Kaffeemaschine, welches kostengünstig ist und am wenigsten Hardware benötigen würde, wird hier nicht mehr genauer betrachtet, da es wie erwähnt rechtliche Probleme mit Jura gäbe und zudem nicht adaptierbar auf andere Systeme wäre.

5.1 Lösungsvorschlag 1 - MDB

Der erste Lösungsvorschlag (Abbildung 8) realisiert ein klassisches bargeldloses Zahlungssystem. Dafür wird das Jura MDB-Interface (Abschnitt 3.2.1) und das MDB2PC Interface benötigt. Das zu realisierende System wird als MDB Cashless Device be-

trieben. Es wird so möglich sein, sich an der Kaffeemaschine mittels einer NFC-Karte oder eines NFC fähigem Mobiltelefon an der Maschine zu authentifizieren und das System kann protokollieren, wer wann was konsumiert hat und dementsprechend Statistiken erstellen. Die funktionale Anforderung der Abrechnung ist somit erfüllt. Es existieren wie bereits erwähnt diverse solcher Systeme auf dem Markt, jedoch sind diese alle sehr teuer. Dieser Systementwurf könnte als Open Source Lösung entwickelt werden. So können kleiner Unternehmen mit wenig Hardware und der frei verfügbaren Software ein einfaches Abrechnungssystem installieren.

Neben dem MDB-Interface und den NFC-Schnittstelle muss noch ein Server entwickelt werden. Dieser Realisiert die Logik und speichert die benutzerspezifischen Informationen. Über den Server sollten dann diverse Statistiken sowie generelle Informationen über die Kaffeemaschine abgerufen werden können.

Die Anforderung der Fehlererkennung und des automatischen Bezugs wird in diesem Entwurf nicht erfüllt.

Die nichtfunktionalen Anforderungen werden von diesem Systementwurf erfüllt. Das System ist dank dem Einsatz von MDB sehr flexibel. Zudem sind die Kosten dieses System tragbar, im Vergleich zu erwerbaren bargeldlosen NFC-Abrechnungssystemen. Die Bedienung kann wie gewünscht umgesetzt werden, da der Lösungsvorschlag die Bedienung nicht beeinträchtigt.

5.2 Lösungsvorschlag 2 - MDB, Robotik

Der zweite Lösungsvorschlag (Abbildung 9) ist eine Erweiterung der Variante 5.1. Neben dem MDB-Interface, NFC-Schnittstelle und dem Server wird in diesem Entwurf noch ein Roboter hinzugefügt. Der Roboter, am besten in Form eines Roboterarm, sollte die automatische Konfiguration und den Bezug des Kaffees realisieren. Der Roboterarm wird über den zentralen Server gesteuert. Falls nun ein Benutzer seine NFC-Karte, mit einem darauf gespeichertem Rezept an den NFC-Reader hält, wird der Roboterarm aufgefordert, den Kaffee für den Benutzer einzustellen und dann zu beziehen. Somit kann die funktionale Anforderung der automatischen Konfiguration und Bezug realisiert werden.

Es ist dabei zu beachten, dass der Roboterarm für jede Kaffeemaschine neu kalibriert werden muss. Jedoch ist das System nach wie vor sehr flexibel. Zudem ist darauf zu achten, dass der Roboter genügend Kraft und Flexibilität aufweist um die Knöpfe zu betätigen. Die Kosten eines leistungsfähigen Roboterarms dürfen nicht unterschätzt werden und erfüllen die Anforderung die Kosten tief zu halten nicht. Die Bedienung der Kaffeemaschine ist massiv eingeschränkt und die Maschine ist manuell nicht mehr oder nur noch schwer bedienbar, da der Roboterarm viel Platz vor der Maschine in Anspruch nimmt. Zudem verzögert sich der Bezug des Kaffees erheblich, da der Roboterarm die Maschine nicht so schnell bedienen kann wie ein Mensch.

Die Anforderung der Verwaltung ist nach wie vor ungelöst.

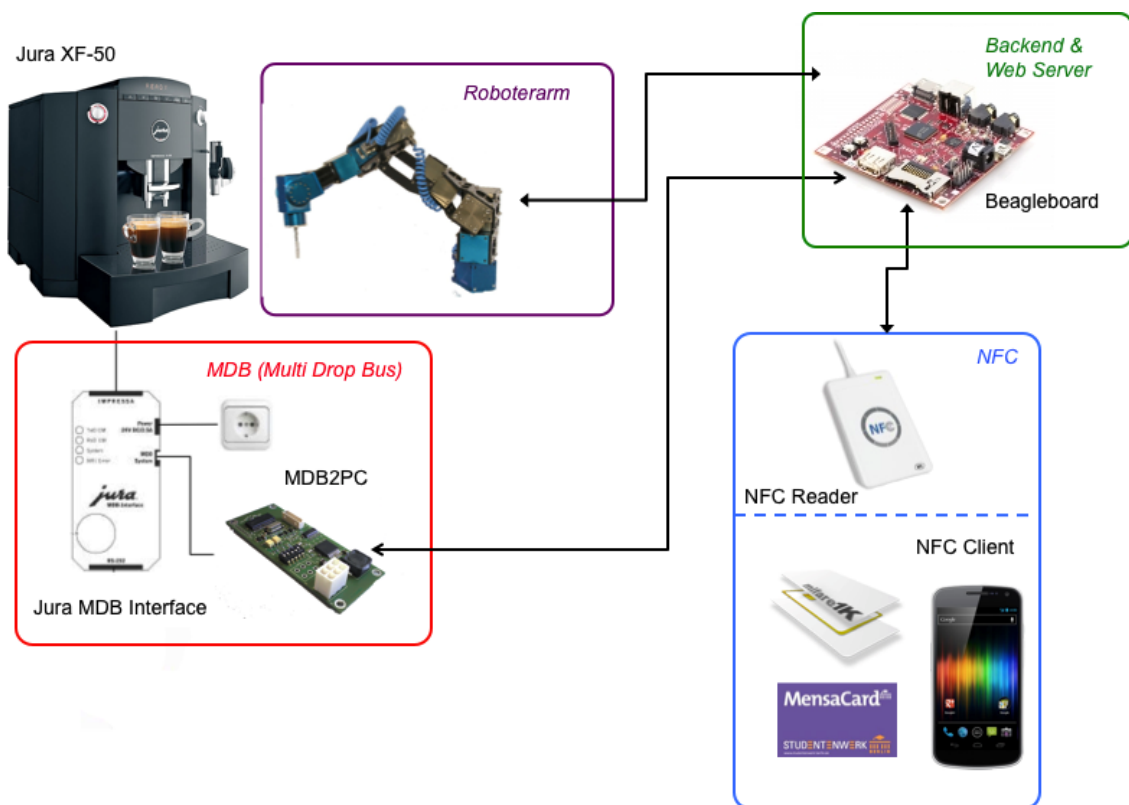


Abbildung 9: Lösungsvorschlag 2 - MDB, Robotik

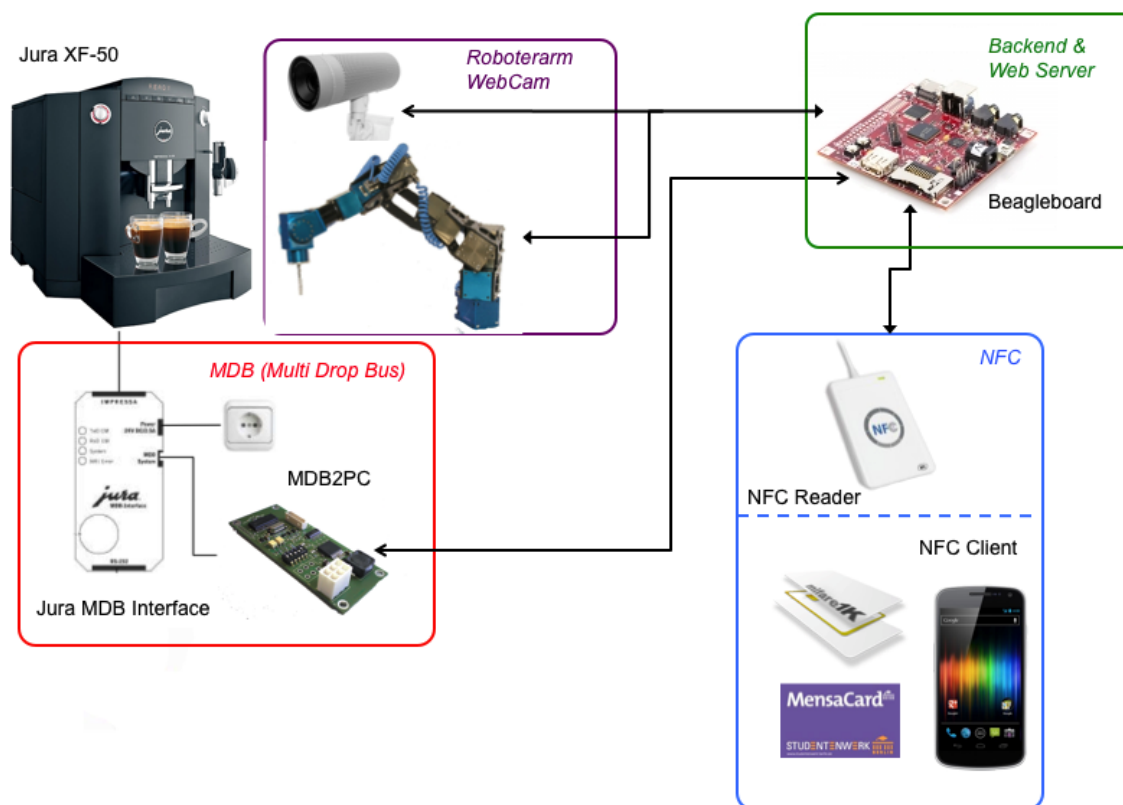


Abbildung 10: Lösungsvorschlag 3 - MDB, Robotik, Kamera

5.3 Lösungsvorschlag 3 - MDB, Robotik, Kamera

Dem dritten Vorschlag (Abbildung 10) wird neben den in Abschnitt 5.2 beschriebenen Bestandteilen noch eine Kamera hinzugefügt. Die Kamera wird direkt auf das Display der Kaffeemaschine ausgerichtet. Die Kamera überwacht nun das Display der Kaffeemaschine und kann die auf dem Display angezeigten Meldungen analysieren. Dafür wird von Server alle paar Sekunden ein Bild des Displays gemacht um den aktuellen Zustand der Maschine zu erfassen. Alle möglichen Meldungen sollte daher erfasst und in einem Zustandsdiagramm gespeichert werden. So kann man jederzeit Fehler und Meldungen der Kaffeemaschine erfassen. Wird nun nach einen Bezug, die Meldung wie z.B. Wasser füllen dargestellt, kann auf dem Server der letzte Benutzer, welcher Kaffee konsumiert hat abgefragt werden und dann durch die in einer Datenbank abgelegten Email Adresse informiert werde, dass er sich doch bitte zur Kaffeemaschine begeben soll um Wasser nachzufüllen. Mit Hilfe der Kamera ist nun auch die Anforderung der Verwaltung der Kaffeemaschine realisiert. In Kombination mit dem MDB-Interface, der NFC-Schnittstelle dem Roboter, der Kamera und dem Server sind nun alle Anforderungen in einem System erfüllt.

Ein erheblicher Nachteil dieses Vorschlags sind die vielen verschiedenen Komponenten, welche eingestellt und angepasst werden müssen. Und die nach wie vor ungelösten Probleme, welche der Roboterarm verursacht.

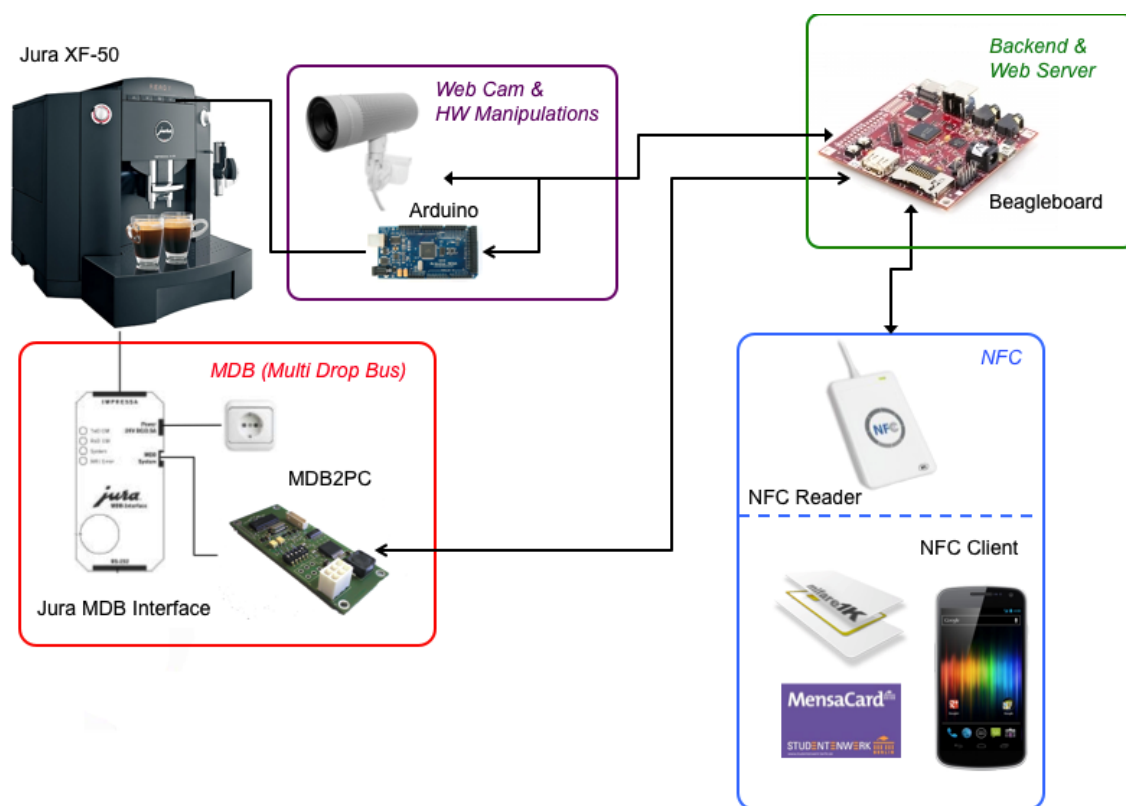


Abbildung 11: Lösungsvorschlag 4 - MDB, HW Manipulation, Kamera

5.4 Lösungsvorschlag 4 - MDB, HW Manipulation, Kamera

Der vierte Lösungsvorschlag (Abbildung 11) ersetzt den Roboterarm mit einer Relaischaltung, welche den automatischen Bezug von bevorzugten Kaffee Rezepten erlaubt. Für diese Installation, müssen Kabel an die Knöpfe der Kaffeemaschine gelötet und mit einem Arduino verbunden werden. So kann über die Kabel mittels einer Relaischaltung ein Knopfdruck simuliert werden (Abschnitt 3.1.2). Hier ist zu beachten, dass man bei den unterschiedlichen Modellen nicht immer gleich guten Zugang zu den Knöpfen der Maschine hat, meistens aber die Möglichkeit hat ein solches Kabel anzulöten und wieder zu entfernen.

In diesem Entwurf sind alle funktionalen Anforderungen gegeben. Die nicht-funktionalen Anforderungen werden auch erfüllt. Das System ist sehr flexibel. Die Bedienung ist nicht beeinträchtigt und die Kosten eines solchen Systems sind gering.

5.5 Lösungsvorschläge Bewertung und Entscheidung

Die verschiedenen Lösungsvorschläge bergen diverse Vor- und Nachteile. Zudem erfüllen sie nicht immer alle Anforderungen. Es ist jedoch wichtig, diese Entwürfe gut miteinander zu vergleichen, damit ein Vorschlag für die Implementierung des Prototyps gewählt werden kann. Die verschiedenen Vorschläge wurden anhand der gestellten Anforderungen verglichen (Tabelle 4).

5 LÖSUNGSVORSCHLÄGE

Jeder Vorschlag wird anhand der gestellten Anforderungen untersucht und bewertet. Die jeweiligen Anforderungen können in einer Skala von 0 bis 10 Punkten bewertet werden. Wobei eine 0, ein nicht erfüllen der Anforderungen bedeutet und eine 10 die Anforderung zur maximalen Zufriedenheit erfüllt.

Anforderung	Lösungs- vorschlag 1	Lösungs- vorschlag 2	Lösungs- vorschlag 3	Lösungs- vorschlag 4
Abrechnung	10	10	10	10
Verwaltung	0	0	8	8
Auto. Bezug	0	6	6	10
Flexibilität	10	8	8	9
Kosten	8	3	3	7
Benutzbarkeit	10	5	4	8
Total	36	32	39	51

Tabelle 4: Lösungsvorschläge Vergleich

Die Lösungsvorschläge 1 und 2 erfüllen nicht alle funktionalen Anforderungen. Lösungsvorschlag 3 und 4 kommen in die genauere Betrachtung. Der Lösungsvorschlag 3 hat deutlich weniger Punkte als Vorschlag 4, da die nichtfunktionalen Anforderungen nur ungenügend erfüllt werden. Das hat damit zu tun, dass einerseits die Bedienung durch den Roboterarm beträchtlich eingeschränkt wird und die Kosten für den Roboterarm sehr hoch sind. Somit bleibt als einziger Lösungsvorschlag der Entwurf 4. Jedoch erreicht auch dieser Vorschlag nicht die maximale Punktzahl. Das hat hauptsächlich mit der Justierung der Kamera und der Anpassung der Filter für die Texterkennung zutun. Trotzdem erreicht dieser Vorschlag eine gute Note und wird deshalb auch für die prototypische Implementierung als Vorlage benutzt.

6 Möglichkeiten der Bedienung

In diesem Abschnitt wird die generelle Interaktion mit dem System analysiert. Es werden die unterschiedlichen Arten, wie mit dem System kommuniziert werden kann und die entsprechenden Vorgänge wie Registrierung und das Verwalten persönlichen Informationen genauer betrachtet.

6.1 Interaktion mit dem System

NFC realisiert eine nicht visuelle Kommunikation. Dies bedeutet, dass der Benutzer nicht sieht ob eine Aktion, wie z.B. das lesen der ID von der NFC-Karte erfolgreich war oder nicht. Es ist jedoch zwingend notwendig dem Benutzer eine Rückmeldung zu geben ob seine gewünschte Transaktion stattgefunden hat oder nicht. Denn es kann durchaus vorkommen, dass die Karte nicht gelesen werden kann, oder der Benutzer zwar Authentifiziert wurde, aber nicht in der Lage ist einen Kaffee zu beziehen weil z.B. sein Kredit für Kaffee aufgebraucht ist oder er vom Administrator gesperrt wurde. Es sind verschiedene Möglichkeiten für die Interaktion mit dem Benutzer denkbar.

So kann man den Benutzer über einen Touchscreen an der Maschine visualisieren, was gerade am System geschieht und ob die Kommunikation zwischen seiner NFC-Karte und dem System funktioniert hat. Der Touchscreen bietet den Vorteil, dass viele und detaillierte Angaben zur Transaktion und generell der Interaktion dargestellt werden können. Nachteilig ist jedoch, dass ein Touchscreen angebracht werden muss und somit Platz direkt an der Kaffeemaschine in Anspruch nimmt.

Eine weitere Variante ist die Information über Transaktionen durch Audiosignale mitzuteilen. So kann man für eine erfolgreiche Interaktion ein helles, freundlich klingendes Signal abspielen. Für eine nicht erfolgreiche Interaktion ein kurzes grell tönendes Signal abspielen. Hier überzeugt der Vorteil, dass ausser kleinen Lautsprecher nichts installiert werden muss, sowie die Realisierung sehr einfach umgesetzt werden kann. Die Nachteile sind jedoch, dass keine detaillierten Informationen über die Transaktion bereitgestellt werden können. Gerade bei einer nicht erfolgreichen Interaktion, ist es wünschenswert den Grund des Misslingens zu erfahren. Ein weiterer Nachteil ist das Interpretieren der Signale. Ist ist nicht immer offensichtlich welches Audiosignal positiv respektive negativ einzustufen ist. Zudem kann der Benutzer keinen Einfluss auf die Transaktion nehmen. Die Kommunikation mit dem System ist mit dieser Variante unidirektionale, da nur das System dem Benutzer minimale Informationen liefert kann und keine Kommunikation von Benutzer zum System möglich ist. Weiter ist zu beachten, dass eine Registrierung direkt an der Maschine nicht möglich ist. Benutzer müssten sich beim Administrator melden, welcher dann die NFC-Karte einem Account zuweist.

Auch denkbar ist ein Text-to-Speech System. Text-to-Speech hat den Vorteil dass die Kommunikation bidirektional sein kann. Es können so auch detaillierte Informationen über die Transaktion dem Benutzer vermittelt werden. Nachteilig hier ist jedoch, dass eine gute Text-to-Speech Implementierung ziemlich aufwändig ist. Zudem ist zu beachten, dass das System sprecherunabhängig ist, sodass Stimmen

von allen beteiligten Benutzer erkannt werden. Jedoch kann man das System so gestalten, dass die Interaktion mit dem System nur über das Bestätigen oder Ablehnen von Aktivitäten realisiert wird. Es ist auch zu beachten, dass ein solches System mehrsprachig gestaltet werden sollte.

Die letzte Interaktionsmöglichkeit, welche in Betracht gezogen wurde, ist jene mit einem Drucker. Hält der Benutzer eine neue nicht registrierte NFC-Karte auf den Reader, wird die ID der Karte ausgedruckt. Mit dieser ID kann sich der Benutzer dann über eine Homepage registrieren oder die neue ID seinem Benutzerkonto hinzufügen. Für die generelle Interaktion müssten dann immer die Informationen ausgedruckt werden. So kann das System nach erfolgreichem Bezug, einen Ausdruck mit der Information des Konsums und dem noch verfügbaren Kredit ausdrucken. Auch dieses System ist nur unidirektionale und zudem unpraktikabel, da ein grosser Wartungsaufwand nötig wäre. Es muss regelmässig Papier und Druckerpatronen nachgefüllt werden.

In der Tabelle 5 sind die verschiedenen Interaktionsmöglichkeiten nochmals aufgelistet.

Kriterien	Touchscreen	Audiosignal	Text-to-Speech	Drucker
Bidirektional	✓	✗	✓	✗
Detail Informationen	✓	✗	✓	✓
Registrierung am System	✓	✗	✓	✗
Wartung	✓	✓	✓	✗

Tabelle 5: Interaktionsmöglichkeiten Vergleich

Wie in Tabelle 5 ersichtlich ist, sind die zwei Varianten Text-to-Speech und der Touchscreen geeignet für die Interaktion mit dem System. Für die Implementierung des Prototyps wurde die Variante mit dem Touchscreen gewählt, da diese Implementierung weniger Zeit in Anspruch nimmt. Es ist jedoch nach wie vor denkbar das System zu einem späteren Zeitpunkt auf eine Interaktion mittels Text-to-Speech umzustellen.

6.2 Bereitstellung der Informationen

Die Informationen des Systems müssen dem Benutzer zugänglich gemacht werden. So sollte der Benutzer möglichst einfach und flexibel auf seine im System gespeicherten Daten zugreifen können. Hier bieten sich grundsätzlich zwei Varianten an. Einerseits können die Informationen direkt an der Kaffeemaschine abgerufen werden, sei es durch den Touchscreen oder der Text-to-Speech Variante. Weiter ist eine Homepage denkbar, auf welcher die Informationen von extern abgerufen werden können. Dass eine Information direkt an der Maschine erforderlich ist, wurde bereits in Ab-

schnitt 6.1 dargestellt. Werden die Benutzerinformation nur an der Maschine direkt dargestellt, muss der Benutzer für jede Änderung seiner persönlichen Informationen an die Kaffeemaschine gehen und diese dort abrufen resp. ändern. Deshalb ist eine Homepage wünschenswert. So können die Benutzer jederzeit und von überall her ihre Informationen abrufen. Zudem wird der Status der Kaffeemaschine erfasst. Eines der Ziele dieser Statuserfassung ist, dass man von extern sehen kann wie der aktuelle Status der Kaffeemaschine ist. Hat man keine Homepage mit welcher man extern den Status abfragen kann, geht eine wichtige Funktion verloren. Es ist folglich ein System (Touchscreen oder Text-to-Speech) wie auch eine Homepage, dementsprechend auch ein Webserver erforderlich.

Neben dem Status der Kaffeemaschine sowie die persönlichen Informationen zu bezogenen Kaffees und erledigten Servicearbeiten, sollten Informationen wie: kürzlich bezogene Kaffees, kürzlich erledigte Servicearbeiten dargestellt werden. Diese Informationen sind nicht benutzerspezifisch, sondern zeigen die benutzerübergreifenden Aktivitäten an der Kaffeemaschine.

6.3 Registrierung

Generell sollte das System einfach zu bedienen sein. Sprich es sollte selbsterklärend sein, damit nicht jeder Benutzer des Systems eine spezielle Schulung benötigt. Auf dem Touchscreen an der Kaffeemaschine sollte deshalb eine einfache und kurze Einleitung ins System bereitgestellt werden. Die Benutzer werden an dem Screen an der Kaffeemaschine aufgefordert eine NFC-Karte an den markierten Reader zu halten. Beim ersten Besuch der Kaffeemaschine wird die NFC-Karte dem System nicht bekannt sein und es muss dem Benutzer die Möglichkeit bieten sich zu registrieren oder diese neue NFC-Karte einem existierenden Konto hinzuzufügen. Bei der Registrierung besteht die Möglichkeit, gleich an der Kaffeemaschine am Touchscreen ein Konto zu eröffnen. Dafür muss der Benutzer Email Adresse, Username und Passwort festlegen. Der Nachteil hier ist, dass der Benutzer viele Informationen über den Touchscreen eingeben muss und solange die Kaffeemaschine blockiert. Eine weitere Möglichkeit wäre, dass der Benutzer nur eine Email Adresse eintippt und sich später über einen Link, welchen er zugeschickt bekommt an seinem Computer registriert. So wäre die Kaffeemaschine nur kurz blockiert. Der Nachteil hier ist jedoch, dass der Benutzer zuerst an seinen Rechner gehen muss um sich zu registrieren und dann wieder zurück an die Maschine gehen muss um einen Kaffee zu beziehen. Um dieses Problem zu beheben, wird dem Benutzer bei der Eingabe seiner Email Adresse einen Kaffee gutgeschrieben. So kann er direkt nach Eingabe seiner Email einen Kaffee konsumieren. Für jeden weiteren Konsum muss sich der Benutzer jedoch zuerst am System komplett registrieren. Wichtig hier ist, dass registrierte NFC-Karten gesperrt bleiben. Damit Benutzer sich nicht neue Emails an unterschiedliche Adressen schicken lassen und somit jedes mal einen Kaffee umsonst beziehen können ohne sich richtig zu registrieren. Es ist nach wie vor denkbar, dass sich Benutzer immer mit neuen NFC-Karten registrieren und somit Kaffee umsonst beziehen können. Jedoch müsste der Benutzer viele solcher NFC-Karten haben.

6.4 Änderung der Daten

Dem Benutzer muss jederzeit die Möglichkeit geboten werden seine Benutzerdaten zu ändern oder gegebenenfalls sein gesamtes Konto zu löschen. Die Änderungen persönlicher Benutzerdaten, sollten nur über die Homepage angeboten werden, denn wenn Benutzer ihre Daten direkt an der Maschine ändern, blockieren sie die Maschine für längere Zeit. Löscht ein Benutzer sein Konto, sollte die NFC-Karte sowie auch die Email für eine gewisse Zeit gesperrt bleiben. So wird verhindert, dass Benutzer kurz vor der Abrechnung von konsumierten Kaffees sein Konto löscht und dann gleich wieder neu anmeldet. Weiter wäre es denkbar Konten nur zu löschen wenn alle Kaffees abgerechnet sind und von Administrator als löschfähig akzeptiert wurden.

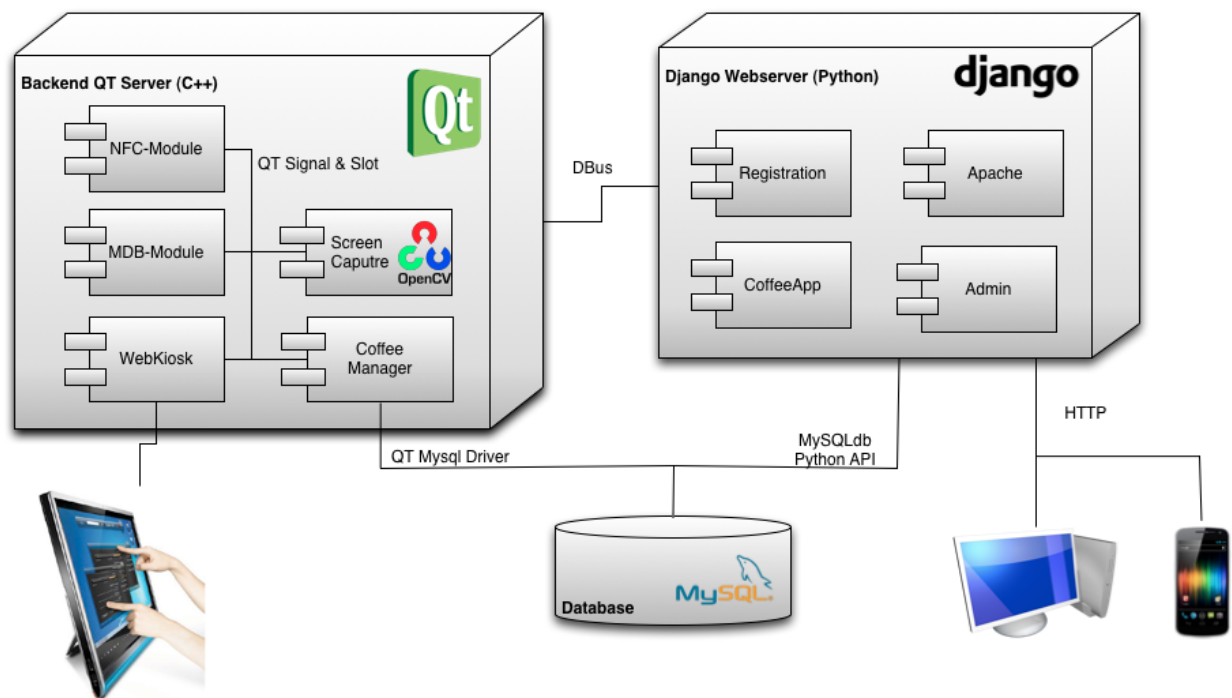


Abbildung 12: Prototyp Systemarchitektur

7 Prototyp Implementierung

Im Rahmen dieses Forschungsprojektes wurde ein Prototyp entwickelt, welcher die Realisierbarkeit der erforschten Systementwürfe demonstriert. Als Vorlage für den Prototypen wurde der Lösungsvorschlag 4 (Abschnitt 5.4) gewählt. Da dieses System sehr umfangreich ist, wurde die Komponente des automatischen Bezugs von Kaffee nicht implementiert. Die Abrechnung mittels MDB, die Authentifizierung mittels NFC, die Verwaltung mittels der Kamera sowie der Webserver mit Homepage wurde im Rahmen dieser Arbeit realisiert. In den folgenden Kapitel werden die verschiedenen Implementierung beschrieben.

7.1 Systemarchitektur

Damit das System als Ganzes funktionieren kann, müssen diverse Module implementiert und zusammengehängt werden. Abbildung 12 visualisiert die Systemarchitektur. Darin sind alle Module und die Kommunikationsprotokolle dargestellt.

Das System besteht aus drei Hauptkomponenten. Der „Kaffeesever“ realisiert die gesamte Funktionalität, welche in direktem Kontakt mit der Kaffeemaschine stehen. Es existieren diverse Module, welche die einzelnen Aufgaben übernehmen. Untereinander kommunizieren sie mittels dem Singal & Slot Konzept von QT. Der Quellcode wurde mit dem Framework QT in C++ geschrieben. Daneben wurde ein Webserver implementiert. Dieser wurde mit dem Web-Framework Django in Python geschrieben. Die dritte Komponente ist die Datenbank, in dieser werden die Daten

wie z.B. Kaffee-Bezug, Servicearbeiten, usw. gespeichert. Sowohl der „Kaffeesever“ als auch der Webserver greifen auf die Datenbank zu. Die Kommunikation zwischen Webserver und dem „Kaffeesever“ wird über das Protokoll D-Bus realisiert. Das ganze System wird auf einem Linux Kubuntu Betriebssystem betrieben.

7.2 Datenspeicherung

Alle Daten die permanent gespeichert werden, sind in einer relationalen MySQL Datenbank abgelegt. Das Datenbankmodell ist in Abbildung 13 abgebildet. Die Datenbank besteht aus acht Tabellen. Zentral ist die Tabelle „auth_user“ hier werden die Informationen des Benutzers gespeichert, wie Username, Passwort, usw. Neben der Benutzer-Tabelle existiert eine „kaffeewebapp_userprofile“ Tabelle welche zusätzliche Informationen über den Benutzer speichert, wie Avatar-Bild, Kaffeekredit, usw. Weiter kann jeder Benutzer mehrere „kaffeewebapp_tellymark“ und „kaffeewebapp_servicelist“ Tabellen besitzen. In diesen werden die konsumierten Kaffees und die erledigten Servicearbeiten des Benutzers gespeichert.

Da sowohl der „Kaffeesever“ als auch der Django Webserver auf die Datenbank zugreifen, sollte ein besonderes Augenmerk auf die „Lost Update“ Anomalie gesetzt werden. Was bedeutet, dass zwei Datenbankoperationen zeitversetzt eine Änderung auf dasselbe Datenobjekt schreiben. D. h., die erste Änderung der beiden geht verloren, da beide Operationen den gleichen Ausgangswert benutzen und nicht die zweite Operation auf das Schreiben der ersten Operation wartet und danach diesen Wert verwendet. Dieses Problem wird vermieden, wenn immer nur eine Applikation Schreibrechte auf die Tabellen hat. So können Tellymarks und Servicearbeiten nur von dem „Kaffeesever“ beschrieben und vom Webserver gelesen werden. Gleiches gilt für die Tabelle „kaffeewebapp_coffeemaschinestate“, auf alle anderen Tabellen hat nur der Webserver Schreibrechte und der „Kaffeesever“ hat auf diese nur Leserechte.

Der Zugriff auf die Datenbank erfolgt bei „Kaffeesever“ mittels einem QT MySQL Plugin, mit welchem zuerst eine Verbindung aufgebaut werden muss. Danach können SQL übliche Kommandos ausgeführt werden. Der Webserver bietet über seine interne Framework-Struktur einen transaktionsgesicherten Zugriff auf die Datenbank.

7.3 Kommunikationsprotokolle

In dem implementierten System kommen diverse Kommunikationsprotokolle zum Einsatz. Die Kommunikation für die Authentifizierung (NFC) wird im Abschnitt 7.4 beschrieben. Die Kommunikation mit der Kaffeemaschine (MDB) wird im Abschnitt 7.5 genauer erläutert. Auf den Webserver wird über das HTTP Protokoll zugegriffen. In den folgenden Unterkapitel werden die Kommunikationsprotokolle Signal & Solt sowie der D-Bus kurz beschrieben.

7 PROTOTYP IMPLEMENTIERUNG

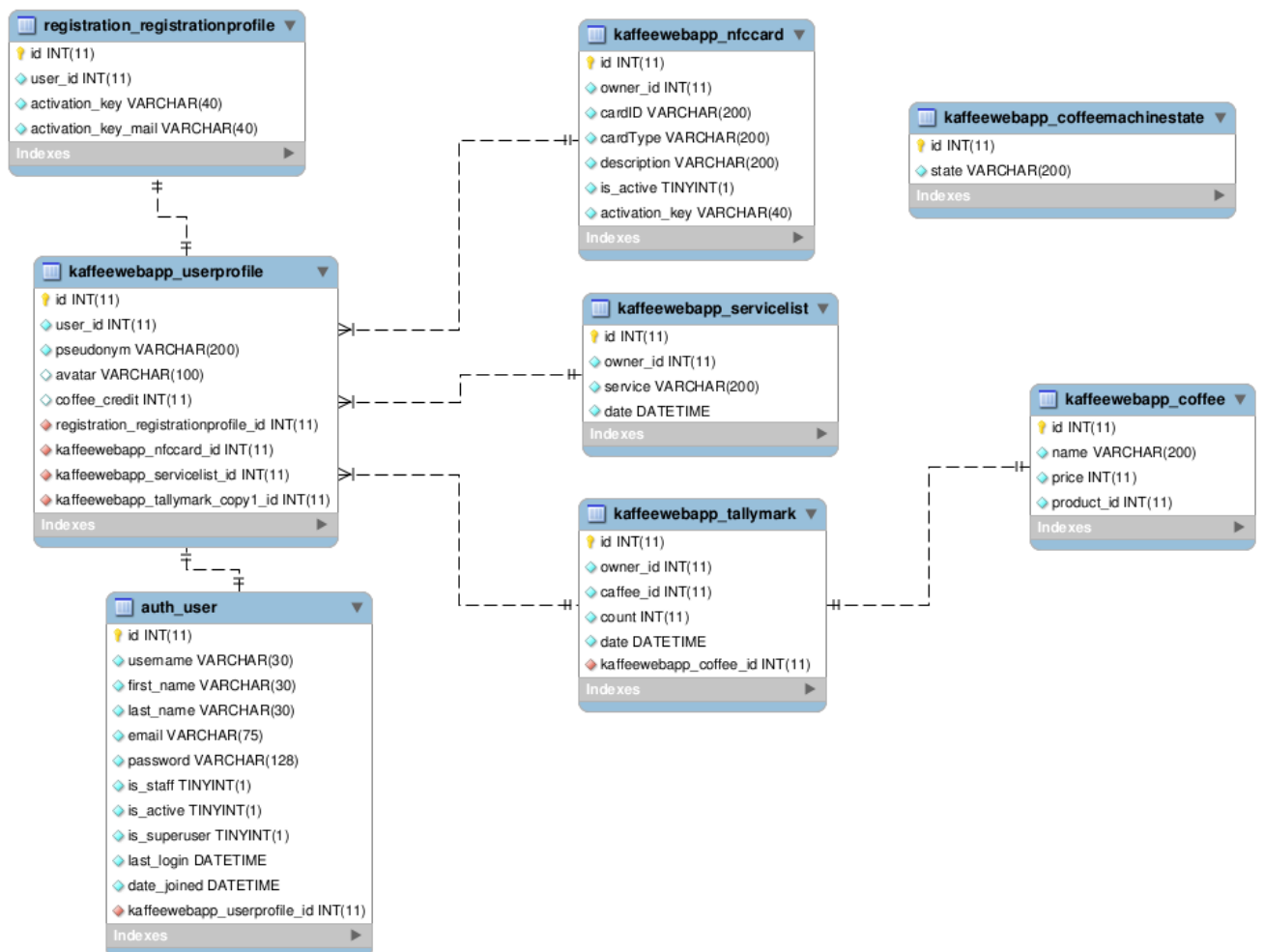


Abbildung 13: Datenbank Diagramm

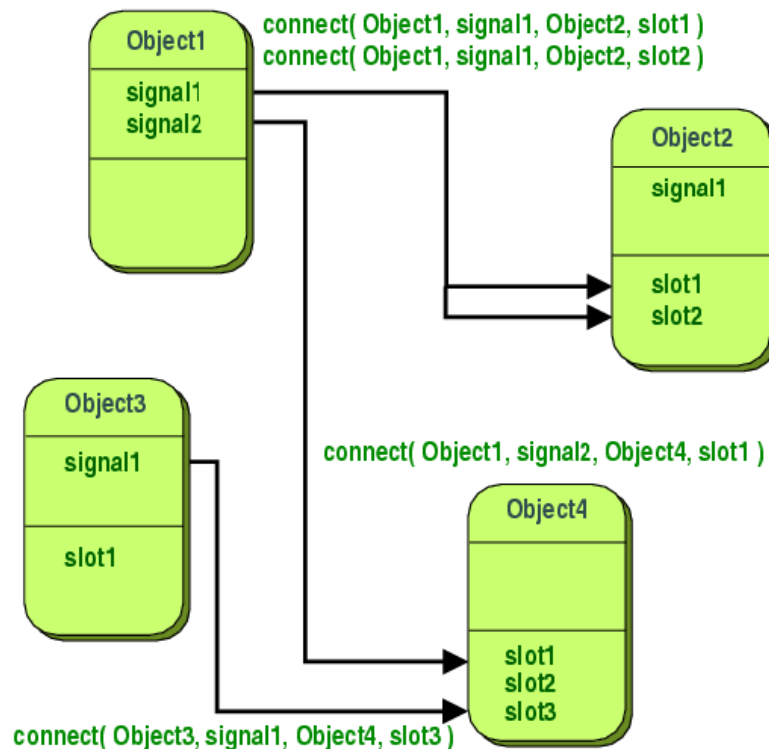


Abbildung 14: QT Signal & Slot Konzept [13]

7.3.1 QT Signal & Slot

Das „Signal & Slot“ Konzept von QT wird benutzt für die Kommunikation zwischen den unterschiedlichen Modulen. Das „Signal & Slot“ realisiert einen ereignisgesteuerten Programmfluss und bietet somit eine Alternative zu Rückruffunktionen (Callbacks). Die dynamische Prüfung der Aufrufparameter bezüglich ihres Typs ist einfacher und flexibler als Callbacks. Signale sind Events, die beim Eintreten eines Ereignisses abgeschickt werden. Ein Slot ist eine gängige Funktion, welche mit einem Signal verknüpft werden kann und auch durch dieses Signal ausgelöst wird. Slots und Signale sind zunächst lose. Erst durch die Verknüpfung der Funktion „connect“ werden Signale den Slots zugeteilt. Bei jedem Auslösen des Signals, wird der verbundene Slot aufgerufen. Ein Signal kann auch mit mehreren Slots verbunden sein, sodass beim Eintreten eines Ereignisses mehrere Funktionen aufgerufen werden. Ebenso kann ein Slot mit mehreren Signalen verbunden werden, wodurch dieselbe Funktion beim Auftreten unterschiedlicher Signale aufgerufen wird. [13] Abbildung 14 visualisiert das QT Signal & Slot Konzept.

7.3.2 D-Bus

D-Bus ist ein Open Source IPC Framework, das sich insbesondere an den Bedürfnissen von Desktopumgebungen einer graphischen Benutzeroberfläche orientiert. Es ist Bestandteil des freedesktop.org Projekts und wird nahezu bei jeder modernen

Linux-Distribution eingesetzt, die über eine graphische Oberfläche verfügt. In seiner Gesamtheit stellt der D-Bus mit all seinen Komponenten einen Object Request Broker (ORB) dar. In diesem Projekt wurde der D-Bus für die Kommunikation zwischen „Kaffeesserver“ und Webserver benutzt.

7.4 NFC Modul

Damit diverse NFC-Karten mit dem System kommunizieren können wird eine NFC-RFID Bibliothek verwendet, welche im Rahmen von Forschungsaktivitäten am Forschungsinstitut INKA entstanden ist. Die Bibliothek abstrahiert die verschiedenen RFID-Reader und Transponder. Es ist so möglich über wenige Kommandos eine ID von einem Transponder auszulesen. Die Bibliothek unterstützt folgende NFC-Reader:

- SCL3711
- TouchATag
- MetraTec

Zudem werden folgende Transponder unterstützt:

- ICode
- MIFARE Desfire
- MIFARE Classic
- MIFARE Ultraligh

Somit können sehr viele unterschiedliche NFC-Karten erkannt und gelesen werden. Von der Mensakarte über NFC-Türschlüssel bis hin zur neuen Gesundheitskarte, alle können durch die Bibliothek mit dem System kommunizieren.

Damit eine ID von einem Transponder gelesen werden kann muss folgender Code umgesetzt werden:

```
1 //Include library
2 #include <nfc/fid/LibnfcDeviceManager.h>
3 //Create a scl3711 device manager
4 LibnfcDeviceManager *libnfcDevice = new LibnfcDeviceManager();
5 //Connect to device
6 libnfcDevice->connect();
7 while{run}{
8     //Get transponders on reader
9     vector<Transponder> transponders = libnfcDevice->readTransponders()
10    ();
11    //Get ID of first Transponder found
12    if(transponders.size()==1){
13        int transponderID = transponders[0].getID();
14        //Inform Kaffeesserver that transponder is found
15        emit transponderFound(transponderID);
16    }
17    // Thread sleep 1 sec
```

```
17 | sleep(1);  
18 | }
```

Listing 1: NFC-RFID Bibliothek Transponder-ID lesen

Auf Zeile 2 wird die Header-Datei der Bibliothek geladen. Danach kann ein NFC-Reader initialisiert werden (Zeile 4). Bevor man ID's lesen kann muss man sich mit dem Reader verbinden (Zeile 6). In einer Schleife werden regelmässig die Transponder, welche sich in Reichweite des Readers befinden gelesen (Zeile 9). Danach muss geprüft werden, ob überhaupt ein Transponder in der Nähe des Readers gefunden wurde (Zeile 11). Angenommen es wurde ein Transponder gefunden, kann wie auf Zeile 12 dargestellt die ID gelesen werden. Weiter wird ein QT Signal ausgelöst, welches das Entdecken des Transponders dem „Kaffeesever“ mitteilt.

7.5 MDB Modul

Leider ist ausser der MDB-Spezifikation keinerlei Information, Dokumentation oder Quellcode für das MDB-Protokoll auffindbar. Deshalb musste der Quellcode für die Kommunikation über das MDB-Protokoll vollständig implementiert werden.

Das MDB-Modul besteht aus zwei Klassen. Einerseits aus einer Abstraktion des seriellen Treibers. Über die serielle Schnittstelle werden die MDB-Kommandos von dem MDB2PC Interface empfangen. Dieses MDB2PC Interface wurde von der Firma Abrantix AG hergestellt. Die MDB-Kommandos werden eingebettet in Frames von der seriellen Schnittstelle empfangen und versendet. Ein Frame ist wie in Listing 2 dargestellt aufgebaut.

```
1 | control-frame ::= STX control-data data DLE ETX
```

Listing 2: Serielles MDB-Frame

Wenn das „control-data“ Byte 0x00 enthält folgt in den „data“ Bytes der MDB-Kommando, welcher vom MDB2PC Interface auf dem MDB-Bus gelesen wurde. Wenn MDB-Kommandos versendet werden sollen muss das „control-data“ Byte auch auf 0x00 gesetzt werden. Dass die MDB-Kommandos aus dem Frame gelesen werden können, muss der Inhalt von den „data“ Bytes herausfiltert und an die MDB Cashless Device Klasse weiterleitet werden. Die MDB Cashless Device Klasse realisiert ein MDB Cashless Device im MDB-Level 2. Dies bedeutet, dass das Zustandsdiagramm (Abbildung 3) vollständig implementiert werden musste. Auf die Auflistung des Quellcodes wird hier verzichtet. Der Quellcode wird aber gerne vom Autor auf Anfrage herausgegeben.

In Abbildung 15 wird ein MDB-Verkaufsablauf visualisiert. Die zu sendenden und empfangenden Kommandos sind in der Abbildung als Hexadezimalwert dargestellt.

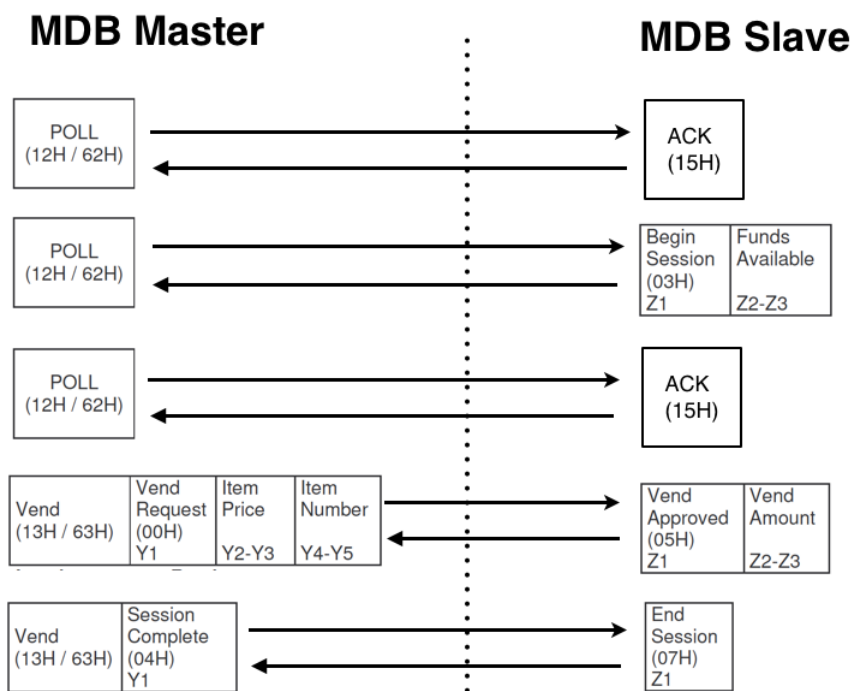


Abbildung 15: MDB Verkauf

7.6 Screen Caputre Modul

Die Texterkennung des Displays der Kaffeemaschine wird in verschiedenen Stufen realisiert. Für das Erkennen wurden zwei Open Source Bibliotheken verwendet. OpenCV wird dafür benutzt um diverse Transformationen und Filter auf dem Kamerabild anzuwenden, sodass danach die eigentliche Texterkennung mittels der Bibliothek Tesseract durchgeführt werden kann.

Das Fotografieren des Displays wird in einem separatem Thread ausgeführt. So werden alle vier Sekunden ein Bild von der Kamera gespeichert. Danach wird das Bild, welches als RGB Bild (Abbildung 16) gespeichert wurde in ein HSV Bild (Abbildung 17) transformiert. So kann dann mittels einem „threshold“ Filter helle, rote Punkte aus dem Bild herausgefiltert werden (Abbildung 18). Da die Texte auf dem Display durch kleine Punkte dargestellt werden, müssen diese mittels einem Filter verbunden werden, sodass sie zu einer Linie übergehen. Dies wird mittels einem „gaussian smoothing“ Filter realisiert (Abbildung 19).

Nun kann das Bild mittels Tesseract analysiert werden. Tesseract liefert den erkannten Text im Bild als String zurück. Leider ist die Texterkennung nicht hundert Prozent zuverlässig und interpretiert gewisse Buchstaben falsch. Damit die exakte Nachricht auf dem Display erkannt werden kann, wird ein Wörterbuch mit möglichen Displaymeldungen erfasst. Der durch Tesseract erkannte Text wird nun mittels einem einfachen Algorithmus mit den jeweiligen Wörtern im Wörterbuch verglichen. Sind mindestens 40 Prozent der Buchstaben an der korrekten Stelle identisch, wird das Wort als erkannt markiert und es wird das Wort aus dem Wörterbuch zurück-



Abbildung 16: Kamera Bild RGB



Abbildung 17: HSV transformiertes Bild



Abbildung 18: Threshold Filter

BEREIT

Abbildung 19: Gaussian smoothing Filter

gegeben.

```
1 bool almostEqual(QString a, QString b){
2     int mismatch=0;
3     if(a.length()>b.length()){
4         for(int i=0; i<b.length();i++){
5             if(a[i]!=b[i]) mismatch++;
6         }
7         double avrError = (double)mismatch/b.length();
8         //qDebug()<< "error marg: " <<QString::number(avrError);
9         if(avrError>0.6) return false;
10        else return true;
11    }else{
12        for(int i=0; i<a.length();i++){
13            if(a[i]!=b[i]) mismatch++;
14        }
15        double avrError =(double) mismatch/a.length();
16        //qDebug()<< "error marg: " <<QString::number(avrError);
17        if(avrError>0.6) return false;
18        else return true;
19    }
20 }
```

Listing 3: Word Matching Algorithmus

7.7 Web Kiosk

Die Gestaltung des Userinterface der graphischen Oberfläche an der Kaffeemaschine, sowie auf der Homepage sollten einheitlich gestaltet werden. Die Anwendung an der Kaffeemaschine kann technisch eine eigenständige Applikation mit Userinterface sein oder ein Fenster, welches mittels einem QT Webview die Homepage darstellt. Der Vorteil einer eigenständigen Applikation ist, dass die Applikation direkt über das QT Signal & Slot Konzept mit den anderen Modulen kommunizieren kann. Dafür muss jedoch eine eigene Applikation entwickelt werden, welche mit der Gestaltung der Homepage identisch sein muss. Dies bedeutet einen deutlichen Mehraufwand für die Realisierung. Werden die Informationen an der Kaffeemaschine direkt über

eine Webview dargestellt ist garantiert, dass das Erscheinungsbild der Applikation identisch ist.

Da ich dieses Projekt alleine umgesetzt habe, wurde die zweite Variante aus Zeitgründen bevorzugt, in welcher an der Kaffeemaschine direkt die Homepage in einer QT Webview dargestellt wird. Die Kommunikation zwischen Homepage und „Kaffeesserver“ wird über den D-Bus realisiert.

7.8 Webserver

Der Webserver wurde mit dem Webframework Django implementiert. Django ist ein in Python geschriebenes quelloffenes Web-Framework, welches dem Model-View-Controller-Schema folgt. Es wurde ursprünglich entwickelt, um die News-Seite „Lawrence Journal World“ zu verwalten und im Juli 2005 unter einer BSD- Lizenz veröffentlicht. Benannt ist es nach dem Jazz-Gitarristen Django Reinhardt. Es wurde im Jahr 2003 als Nonprofit-Projekt gegründet und steht seit 2008 unter der Marke der Django-Foundation.

Ein Django-Projekt besteht aus einer bis beliebig vielen Apps. Diese Apps können kombiniert werden. In Apps werden Funktionalitäten gekapselt, um sie an anderer Stelle wieder verwenden zu können. Das MVC-Konzept ist in Django etwas speziell umgesetzt. Es wird hier als MTV - Model Template View beschrieben. „View“ bezeichnet hier die Auswahl der Daten die der Benutzer zu sehen bekommt, aber nicht unbedingt wie diese Daten aussehen. Die View ist eine in Python geschriebene Callback-Funktion für eine bestimmte URL. Die View delegiert die Präsentation der Daten an ein Template. Templates sind in Djangos mächtiger Template-Sprache geschrieben und können voneinander abgeleitet werden. Es können Blöcke definiert werden, welche von abgeleiteten Templates beschrieben werden. Auf die in Views definierten Variablen kann zugegriffen werden. Einige Abfragen können direkt aus dem Template generiert werden, sowie Fallunterscheidungen getroffen werden und vieles mehr. Das Model steckt in der Definition der Datenbank und ist analog zum Model aus MVC Frameworks zu verstehen. [12]

Es wurden diverse Apps für die Realisierung benutzt. In der „CoffeeApp“ (Abbildung 12) werden alle Anfragen welche Kaffees oder Servicearbeiten betreffen behandelt. So wird durch diese App dem Benutzer seinen Kaffee-Konsum oder seine Servicearbeiten bereitgestellt. Das App „Registration“ erledigt der gesamte Anmeldeprozess. Diese App wird von Django angeboten und wurde für dieses Projekt angepasst. Weiter verfügt der Webserver über eine Admin App, in welcher Administratoren Daten erfassen und löschen können.

8 Fazit

Dieses Forschungsprojekt demonstriert die Möglichkeiten von NFC in Kombination mit der Verwaltung und Bedienung von Kaffeemaschinen und generellen Haushaltsgeräten. Die in diesem Forschungsprojekt erarbeiteten Lösungsvorschläge können auf alle Geräte angewendet werden, welche über ein MDB-Interface verfügen. Somit ist das System sehr flexibel und kann in unterschiedlichen Umgebungen angewendet werden. Zudem wird sehr schön demonstriert, wie man existierende NFC-Karten wieder verwenden kann. Die meisten Systeme welche mit NFC eine Abrechnung realisieren, benötigen eigene NFC-Karten für ihr System. Hier werden schon existierende NFC-Karten wiederverwendet. So ist es nicht nötig, eigene NFC-Karte den jeweiligen Benutzer herauszugeben. Zudem können mehrere unterschiedliche NFC-Karten für die Interaktion mit dem System verwendet werden. Das macht das erforschte System und die prototypische Implementierung sehr flexibel und zeigt das Potenzial von NFC in Kombination mit MDB.

Zudem ist das erforschte System sehr kostengünstig im Vergleich zu erwerblichen NFC-Abrechnungssystemen. Der gesamte Quellcode, welcher im Rahmen der Implementierung des Prototyps realisiert wurde ist komplett Open Source. So können kleinere Unternehmen diesen Quellcode benutzen um ein eigenes Abrechnungs- und Verwaltungssystem zu realisieren. Die MDB-Hardware muss jedoch zwingend selbst hergestellt oder von einem Unternehmen bezogen werden. Das implementierte System kann auf jedem aktuellen Linux Betriebssystem installiert und betrieben werden.

Der Prototyp zeigt sehr schön, dass die erforschten Lösungsansätze einsatzfähig sind. Das entwickelte System ermöglicht eine zuverlässige Verwaltung und Abrechnung eines Kaffeefullautomaten.

8.1 Ausblick

Dieses Projekt wird über das Forschungsprojekt noch weitergeführt. Für den automatischen Bezug und die Konfiguration von Kaffee werden die Knöpfe an der Maschine mit Kabel verlötet. Somit wäre auch die letzte funktionale Anforderung im Prototyp umgesetzt. Weiter wird eine Applikation für Mobiltelefone entstehen, über welche man Kaffee konfigurieren und beziehen kann.

Es existieren bereits diverse Ideen funktionaler Erweiterungen des System. So wäre ein Roboter denkbar, welcher Kaffee servieren kann. Dies bedeutet, dass man den Roboter an seinen Arbeitsplatz bestellen und dann auf ein Tablett, welches am Roboter befestigt ist seine Tasse stellen kann. Der Roboter fährt dann eigenständig an die Kaffeemaschine, bezieht den vorkonfigurierten Kaffee automatisch und fährt diesen zurück an den Arbeitsplatz.

Weiter könnte man das System an die Open Source Währung „Bitcoin“ anschließen. Sodass die Abrechnung direkt in „Bitcoins“ realisiert wird.

Falls ein Benutzer keine NFC-Karte bei sich trägt, wäre eine Erweiterung des System denkbar, sodass Kaffee frei programmiert oder gespielt werden kann. So würde der Benutzer aufgefordert auf dem Touchscreen an der Kaffeemaschine eine

Programmieraufgabe zu erledigen, ist diese erfolgreich gelöst könne er einen Kaffee umsonst beziehen. Gleiches Prinzip ist denkbar mit kleinen Strategie-Spielen, welche direkt am Touchscreen gespielt werden könnten.

Es ist zudem denkbar und erstrebenswert, das ganze System mit künstlicher Intelligenz für die Profilbildung und den automatischen Bezug von Kaffee zu erweitern. So kann das System benutzerspezifische Profile erstellen. Dabei wird der Konsum des Benutzers analysiert und daraus werden Kaffeevorschläge berechnet. Wenn z.B. ein Benutzer bis 10 Uhr morgens immer Kaffee trinkt und für die restliche Zeit des Tages nur noch Espresso, kann die Maschine dem Benutzer zeitabhängig den gewünschten Kaffee automatisch herauslassen.

Zudem ist eine Logik denkbar, welche die konsumierten Kaffees und geleisteten Servicearbeiten kombiniert und daraus ein Profil berechnet. Diese Profile können dann für den Kauf von Kaffeebohnen berücksichtigt werden. So muss jemand der viel Kaffee konsumiert keine Bohnen kaufen, wenn er im Gegenzug viele Servicearbeiten an der Maschine durchführt.

Literatur

- [1] ISO/IEC 18092 Information technology — Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol. (04.2004), Nr. ISO/IEC 18092:2004(E)
- [2] *MDB2PC, MDB to Serial converter*. <http://www.abrantix.com/en-mdb2pc.html>. Version: 06 2012, Abruf: 15.09.2012
- [3] *Textpresso by Zipwhip - Bildquelle*. <http://img.kapook.com/u/tawutchai/Textpresso/Espressp.jpg>. Version: 05 2012, Abruf: 10.09.2012
- [4] *Zipwhip Powers World's First Text-Enabled Espresso Machine*. <http://blog.zipwhip.com/2012/04/30/textpresso-machine-celebrates-cloud-texting-technology/>. Version: 04 2012, Abruf: 17.09.2012
- [5] AG, JURA E.: MDB-Interface Bedienungsanleitung. (08.2007), Nr. 67653
- [6] ASSOCIATION, National Automatic M.: Multi-Drop Bus / Internal Communication Protocol Version 4.0. (04.2009)
- [7] ASSOCIATION, National Automatic M.: Multi-Drop Bus / Internal Communication Protocol Version 1.0. (10.1993)
- [8] COMPANY, Abrantix: the s.: MDB2PC Specification. (2012)
- [9] FINKENZELLER, Klaus: *RFID-Handbuch: Grundlagen und praktische Anwendungen von Transpondern, kontaktlosen Chipkarten und NFC*. Carl Hanser Verlag GmbH CO. KG, 2012
- [10] GERHARD AUSTALLER, Jussi K.: Applications for Smart Environments. (07.2006), Nr. TUD-CS-2006-4
- [11] INC., ProtectStar™: ImPressa Z5 & ImPressa WeB PiLoT Innovation Award. (2006)
- [12] JACOB KAPLAN-MOSS, Adrian H.: *The Definitive Guide to Django: Web Development Done Right*. Apress, 10.2007
- [13] JASMIN BLANCHETTE, SUMMERFIELD M.: *C++ GUI Programming with QT 4*. Second Edition. Trolltech., 2008
- [14] JOSEF LANGER, Michael R.: *Anwendungen und Technik von Near Field Communication (NFC)*. Springer, 2010
- [15] MARCUS STÄNDER, Niklas Lochschmidt Christian Klos Bastian Renner Max M. Aristotelis Hadjakos: A Smart Kitchen Infrastructure. (09.2012)

LITERATUR

- [16] MEYN, Hauke: ISO/IEC 14443-4:2001 Identification cards - Contactless integrated circuit(s) cards - Proximity cards -. (2001)
- [17] ONDRUS, J.: An Assessment of NFC for Future Mobile Payment Systems. (07.2007), Nr. 0-7695-2803-1
- [18] STAFFORD-FRASER, Quentin: *The Trojan Room Coffee Pot*. <http://www.cl.cam.ac.uk/coffee/qsf/coffee.html>. Version: 05 1995, Abruf: 20.09.2012