Exposé

# Thema: „Entwicklung von intelligenten Türschildern auf der Basis von IoT in einem dezentralen Netzwerk“

Von Tolgay Usul

Das Thema „Internet of Things“ (kurz: IoT) ist ein zukunftsträchtiges Thema. Das Bestreben von Vernetzungen zwischen physischen und virtuellen Gegenständen soll die Informationsverarbeitung, sowie den Informationsaustausch vereinfachen. Beispielsweise können dadurch bestimmte Sensordaten durch das einfache Scannen eines QR-Codes ausgelesen werden. Die eigentliche Verarbeitung passiert dabei im Hintergrund.

In dieser Masterarbeit soll ein IoT System implementiert werden. Mit der Eingabe von Daten soll die implementierte Software in der Lage sein mehrere Profile, Ankündigungen, einen Belegungsplan, sowie Veranstaltungen übersichtlich anzuzeigen. Die Eingaben sollen dabei schlicht gehalten werden. Es sollen keine Vorkenntnisse für die Benutzung der Anwendung notwendig sein. Die Installation der Anwendung auf dem IoT Gerät und auf dem Computer werden auf ein Minimum beschränkt. Das schließt die Verwendung von zusätzlicher Software wie einer Datenbank aus. Das Ziel ist es eine funktionsfähige und anwenderfreundliche Applikation zu erstellen.

Für die Implementierung muss zunächst die passende Hardware ausgewählt werden. Es werden drei verschiedene IoT Boards betrachtet und für eine mögliche Nutzung ausgewertet. Zur Auswahl stehen der „Uno Rev 3“ von Arduino, das „Esp-Wrover-Kit-VB“ von Expressif und der „Raspberry Pi 3“ von der Raspberry Pi Foundation. Die oben genannten IoT- Boards sind die derzeit aktuellsten Artikel auf dem Markt. Jedes IoT-Board hat besondere Spezifikationen, die sie voneinander unterscheidet. Das Esp-Wrover-Kit-VB besitzt ein eingebautes LCD, Bluetooth und WiFi. Der Uno Rev 3 ist der günstigste von allen Boards und gilt als Starter Board für die Anwendungsentwicklungen. Ein Display ist nicht vorhanden und muss separat angeschafft werden. Das Raspberry Pi 3 ist das wohl das bekannteste IoT-Board. Es hat einen HDMI-Anschluss, WiFi, Bluetooth und einen Ethernet Anschluss. Das Display muss auch hier separat angeschafft werden. Wegen des HDMI-Anschlusses ist die Anschaffung des Displays ohne besondere Vorgaben möglich. Für die Realisierung des Projektes wird ein Display und mindestens ein Ethernet Anschluss benötigt. Die IoT Boards werden auf Leistung, Preis, Wartung und Nutzungsumfeld miteinander verglichen. Am Ende des Vergleiches soll die Hardware, die die oben genannten Punkte bestmöglich erfüllt, für die Entwicklung der Anwendung genutzt werden.

Nach der Wahl der passenden Hardware, ist es notwendig ein passendes Betriebssystem auszuwählen. Das Betriebssystem muss sowohl stabil sein, als auch IoT Funktionalität erfüllen. Die Installation von Anwendungen soll möglichst einfach und extern ablaufen können, um eine Wartung zu vereinfachen. Das System sollte zudem frei verfügbar sein. Als Betriebssysteme werden Windows 10 IoT, die Standard Linux-Distribution von Raspberry Pi und das neue Android Things miteinander verglichen. Es handelt sich dabei um die drei bekanntesten IoT-Betriebssysteme, die auf dem Markt sind.

Das gemeinsame Netzwerk, wo die IoT-Geräte mit dem Computer kommunizieren sollen, wird ein dezentrales Netzwerk sein. Ein dezentrales Netzwerk hat gegenüber einem zentralen Netzwerk den Vorteil, dass durch das Wegfallen eines Knotenpunktes im Netz das System weiterhin intakt ist. Wenn das Gesamtnetzwerk ausfällt, bilden die einzelnen Knotenpunkte kleinere Netze. Die Funktionalität bleibt erhalten. Nur der Austausch von Daten ist bis zur Wiederherstellung auf die einzelnen Netze beschränkt. Bei einem zentralen Netz würde der Ausfall des zentralen Knotens, die Auflösung des Netzwerkes bedeuten.

Vorläufige Gliederung

1. Einleitung
   1. Motivation
   2. Ziel
   3. Aufbau der Arbeit
2. Grundlagen
   1. IoT Hardware

2.1.1. Uno Rev 3

2.1.2. Esp-Wrover-Kit-VB

2.1.3. Raspberry Pi 3

* 1. Vernetzungen
     1. Zentrale Vernetzung
     2. Dezentrale Vernetzung
     3. Distribuierte Vernetzung
  2. IoT Betriebssysteme
     1. Windows 10 IoT
     2. Raspberry Pi Linux-Distribution
     3. Android Things
     4. Weitere IoT Systeme
  3. REST und SOAP
     1. REST
     2. SOAP
     3. Der Vergleich
  4. Sprachen für die Implementierung grafischer Oberflächen
     1. HTML
     2. XAML
  5. Die Programmierumgebung
     1. Java
     2. C#
     3. Visual Studio 2017
     4. Universelle Windows Plattform (UWP)
     5. Windows Runtime

1. Anforderungen und Zielsetzung
   1. Anforderungsanalyse
   2. Konzept
2. **Anwendungsentwurf und -implementierung**
   1. Architektur der Anwendung
   2. Frameworks und Erweiterungen
   3. Serverseitige Implementierung (Architektur)
      1. Die REST-Schnittstelle (Background-APP)
      2. Das Layout der Anwendung (Foreground-APP)
      3. Der Nachrichtenkanal (UWP-MessageRelay)
      4. Die Prüfungsapplikation (ForegroundPrüfungs-APP)
   4. Clientseitige Implementierung (Architektur)
      1. Layout der Anwendung
      2. Implementierung der Anwendung
      3. Validierung der einzelnen Felder
      4. Speicher- und Profilsystem
3. Benutzerhandbuch
4. Zusammenfassung
   1. Fazit
   2. Ausblick

**Abbildungsverzeichnis**

**Algorithmenverzeichnis**

**Literaturverzeichnis**

**Erklärung**

# Vorläufiges Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | C. Floerkemeier, L. Marc, E. Fleisch, F. Mattern und S. Sarma, The Internet of Things, Zürich, Schweiz: Springer, 2008. |
| [2] | J. Likeness und J. Garland, Programming the Windows Runtime by Example, North Carolina, USA: Addison-Wesley, 2014. |
| [3] | M. Steier, H. Schwichtenberg, Fischer, Mathias und J. Krause, Verteilte Systeme und Services mit .NET 4.5, München: Carl Hanser Verlag GmbH Co KG, 2013. |
| [4] | Microsoft, „Windows Dev Center,“ Oktober 2018. [Online]. Available: https://developer.microsoft.com/de-de/windows/iot. |
| [5] | Google, „Android Things,“ 26 Oktober 2018. [Online]. Available: https://developer.android.com/. |
| [6] | Raspberry Pi Foundation, „Raspberry Pi,“ 26 Oktober 2018. [Online]. Available: https://www.raspberrypi.org/education/. |