

TITEL

Seminararbeit

Version 0.01

Funktionale Programmierung
Zürcher Hochschule für Angewandte Wissenschaften

Simon Lang, Daniel Brun

Date

Versionshistorie

Version	Datum	Autor(en)	Änderungen
0.01	13.04.2015	DBRU	Initiale Version

Daniel Brun (DBRU)

Abstract

Ausgangslage und Ziel

Vorgehensweise

Detaillkonzept & Proof-of-Concept

Eigenständigkeitserklärung

Hiermit bestätige ich, dass vorliegende Semesterarbeit zum Thema „BigData mit RaspberryPi und F#“ gemäss freigegebener Aufgabenstellung ohne jede fremde Hilfe und unter Benutzung der angegebenen Quellen im Rahmen der gültigen Reglemente selbständig verfasst wurde.

Zürich, 24.09.2015

Simon Lang, Daniel Brun

Inhaltsverzeichnis

1	Einleitung	1
1.1	Hintergrund	1
1.2	Ziel	1
1.3	Aufgabenstellung	2
1.4	Erwartete Resultate	2
1.5	Abgrenzung	2
1.6	Motivation	2
1.7	Struktur	2
1.8	Planung	2
2	Recherche	3
2.1	Ausgangslage	3
2.2	Datenerhebung mit dem Raspberry Pi	3
2.2.1	Raspbian mit Mono	4
3	Analyse	5
4	Schlusswort	6
4.1	Fazit	6
4.2	Vergleich: Ist -/ Soll-Planung	6
4.3	Dank	6
	Quellenverzeichnis	8
	Anhang	10
A	Anhang	10

KAPITEL 1

Einleitung

Diese Arbeit wurde als Seminararbeit zur Vorlesung von funktionalen Programmiersprachen verfasst. In diesem Kapitel wird die Aufgabenstellungen und Rahmenbedingungen der Arbeit erläutert.

1.1 Hintergrund

Einer immer grösseren Beliebtheit erfreuen sich kleine Alltagsgegenstände welche mit dem Internet verbunden sind. Dieser Bereich wird IoT genannt. Diese Gegenstände sind in der Lage Daten zu erheben und weiterzuleiten. Da es zukünftig voraussichtlich immer mehr IoT Gegenstände geben wird fallen immer mehr Daten an. Diese Daten werden wegen ihrer Masse auch BigData genannt.

In dieser Arbeit wird evaluiert wie sich funktionale Programmiersprachen im Bezug auf IoT eignen, um BigData auszuwerten.

1.2 Ziel

Mit einem IoT Gerät sollen Daten aufgezeichnet werden. Diese werden als BigData gesammelt und sollen mit einer funktionalen Programmiersprache ausgewertet und ansprechend ausgegeben werden.

Das Hauptziel der Arbeit besteht darin zu überprüfen wie geeignet funktionale Programmiersprachen für die Auswertung von BigData sind. Als Nebenziel soll evaluiert werden, ob eine funktionale Programmiersprache zum erfassen von Daten auf einem IoT Gerät verwendet werden kann.

1.3 Aufgabenstellung

Die freigegebene Aufgabenstellung lautet wie folgt:

- Projektname: Seminar BigData mit RaspberryPi und F#
- Ausgangslage: Durch die rasante Entwicklung im Bereich IoT ergeben sich viele neue Anwendungsmöglichkeiten. Da der RaspberryPI immer leistungsfähiger geworden ist, soll evaluiert werden ob er sich für den Einsatz von funktionalen Sprachen im Bereich BigData eignet.
- Ziel der Arbeit: Es soll gezeigt werden wie F# Sharp auf einem RaspberryPi im Bereich BigData und IoT eingesetzt werden kann.
- Aufgabenstellung: Es soll gezeigt werden, wie eine funktionale Programmiersprache (F#) im Kontext von BigData und IoT eingesetzt und verwendet werden kann. Es soll eine Anwendung zur Sammlung von Sensordaten auf einem Raspberry PI und eine Anwendung zur Analyse / Auswertung der gesammelten Daten implementiert werden.

1.4 Erwartete Resultate

Gemäss freigegebener Aufgabenstellung werden folgende Resultate erwartet:

- Dokumentation
- Implementation / Prototyp

1.5 Abgrenzung

Aufgrund des Umfanges der Arbeit und der begrenzten Zeitdauer werden folgende Punkte von der Arbeit abgegrenzt:

- **Schnittstellendokumentation**
In dieser Arbeit werden nicht die Schnittstellendokumentationen und -spezifikationen rekonstruiert. Es werden jeweils die relevanten Aspekte betrachtet und hervorgehoben.

1.6 Motivation

1.7 Struktur

1.8 Planung

KAPITEL 2

Recherche

In diesem Kapitel werden die Grundlagen recherchiert wie die beiden Teilprojekte, BigData erheben und auswerten, angegangen werden können.

2.1 Ausgangslage

Die Vorlesung zu diesem Semniar befasst sich mit der Programmiersprache F#, weshalb auch dieses Seminar mit derselben Sprache umgesetzt wird. Als IoT Gerät wird ein Raspberry Pi¹ verwendet. Dies ist ein Einplatinencomputer welches von der britischen Raspberry Pi Foundation entwickelt wurde. Der Vorteil daran ist, dass es weit verbreitet ist² und viele Sensoren auf dem Markt gibt mit welchem man Daten sammeln kann³.

2.2 F# mit dem Raspberry Pi

Am 29 Februar 2016 ist das neue Model Raspberry Pi 3 auf dem Markt erschienen⁴. Einige Zahlen zu dem Gerät:

- 1.2GHz 64-bit quad-core ARM Cortex-A53 CPU (10x die Leistung eines Raspberry Pi 1 und 50-60% die Leistung eines Raspberry Pi 2)
- Integriertes 802.11n wireless LAN und Bluetooth 4.1
- Komplette Kompatibilität zu dem Raspberry Pi 1 und 2

Um F# auf dem Gerät laufen zu lassen, gibt es zwei Möglichkeiten, welche nachfolgend erläutert werden.

1 [Rasb.](#)

2 [Rasd.](#)

3 [Rasa.](#)

4 [Rasc.](#)

2.2.1 Raspbian mit Mono

Ein weit verbreitetes Betriebssystem für das Raspberry Pi ist das Raspbian¹. Bei dem Namen handelt es sich um eine Zusammenfassung von Raspberry und Debian. Demnach handelt es sich auch um eine Debian Distribution.

Um F# auf dem Raspbian laufen zu lassen wird Mono benötigt². Dabei handelt es sich um eine Open Source implementierung von Microsoft's .NET Framework, welches für die Ausführung von F# benötigt wird.

2.2.2 Windows 10 IoT

Microsoft hat mit Windows 10 IoT eine Version ihres Betriebssystem herausgebracht, welches speziell für leistungsschwächere Geräte entwickelt wurde³. Bei der IoT Version von Windows 10 ist ein .NET Framework bereits vorhanden. Demnach sollte es keine Probleme darstellen F# darauf laufen zu lassen.

2.3 Sensoren

Für den Raspberry Pi gibt es viele Sensoren auf dem Markt. Herauskristallisiert hat sich jedoch das Starter Kit GrovePi+⁴. Dieses kommt mit einem Ton-, Temperatur-, Feuchtigkeit-, Lichts- und noch weiteren Sensoren. Vorteil an dem Kit ist es, dass es eine Library für .NET gibt mit welchem die Sensoren angesprochen werden können⁵.

1 [Fro.](#)

2 [Mon.](#)

3 [Win.](#)

4 [Gro.](#)

5 [Nug.](#)

KAPITEL 3

Analyse

KAPITEL 4

Schlusswort

4.1 Fazit

4.2 Vergleich: Ist -/ Soll-Planung

4.3 Dank

Akronyme

Bezeichnung	Beschreibung
ERP	Enterprise-Resource-Planning

Glossar

IoT

Unter Internet of Things (IoT) versteht man Alltagsgegenstände, welche mit dem Internet verbunden sind. Im Gegensatz zu normalen Computern, mit denen man sich aktiv beschäftigt, soll IoT den Menschen unterstützen in seinem Tun, ohne dass dieser etwas davon merkt.

Quellenverzeichnis

- [Win] *Develop Windows 10 IoT apps on Raspberry Pi 3 and Arduino - Windows IoT.* URL: <https://developer.microsoft.com/en-us/windows/iot> (besucht am 24. 04. 2016).
- [Fro] *FrontPage - Raspbian.* URL: <https://www.raspbian.org/> (besucht am 24. 04. 2016) (siehe S. 4).
- [Mon] *Home / Mono.* URL: <http://www.mono-project.com/> (besucht am 24. 04. 2016) (siehe S. 4).
- [Nug] *NuGet Gallery / GrovePi for Windows IoT 1.0.6.* URL: <https://www.nuget.org/packages/GrovePi/> (besucht am 24. 04. 2016).
- [Rasa] *Raspberry Pi - Sensor.* URL: <https://www.pi-shop.ch/sensoren> (besucht am 24. 04. 2016) (siehe S. 3).
- [Rasb] *Raspberry Pi - Teach, Learn, and Make with Raspberry Pi.* URL: <https://www.raspberrypi.org/> (besucht am 24. 04. 2016) (siehe S. 3).
- [Rasc] *Raspberry Pi 3 on sale now at \$35 - Raspberry Pi.* URL: <https://www.raspberrypi.org/blog/raspberry-pi-3-on-sale/> (besucht am 24. 04. 2016) (siehe S. 3).
- [Rasd] *Raspberry Pi: Die Erfolgsgeschichte des Scheckkarten-Computers / Gründerszene.* URL: <http://www.gruenderszene.de/allgemein/raspberry-pi-eben-upton> (besucht am 24. 04. 2016) (siehe S. 3).
- [Gro] *Raspberry Pi Internet of Things Kit.* URL: <http://www.dexterindustries.com/shop/grovepi-starter-kit-2/> (besucht am 24. 04. 2016).

Abbildungsverzeichnis

ANHANG A

Anhang
