

Institut für Informatik

Bachelorarbeit

Indoor Positionierung mittels Bluetooth Low Energy

Kevin Seidel

24.09.2013

Erstgutachter: Prof. Dr. Oliver Vornberger

Zweitgutachterin: Prof. Dr. Elke Pulvermüller

Danksagungen

Hiermit möchte ich allen Personen danken, die mich bei der Erstellung der Arbeit unterstützt haben:

- Herrn Prof. Dr. Oliver Vornberger für die Tätigkeit als Erstgutachter und für die Bereitstellung der interessanten Thematik.
- Frau Prof. Dr. Elke Pulvermüller, die sich als Zweitgutachterin zur Verfügung gestellt hat.

Zusammenfassung

Bluetooth

Abstract

Bluetooth

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Ziele der Bachelorarbeit	2
2	Technologien	3
2.1	Bluetooth 4.0	3
2.2	Bluetooth Low Energy	3
2.2.1	iBeacons	4
2.3	iOS, OS X und Xcode	5
2.4	CoreLocation-Framework	5
2.4.1	iBeacons-API	5
2.4.2	Weitere API's	5
2.5	CoreData-Framework	5
3	Werkzeuge	7
3.1	Xcode	7
3.2	Objective C	8
3.3	Versionsverwaltung mit Git	8
3.4	iOS Developer Program	8
3.5	iPhone	8
4	Daten und Messungen	11
4.1	Mobile iBeacons	11
4.2	Stationäre iBeacons	11
4.3	Außenmessungen	11
4.4	Innenraummessungen	11
5	Umsetzung und Implementation	13
5.1	Ansatz zur Positionsbestimmung	13
5.2	Trilateration	13
5.3	Fingerprinting	13
6	Fingerprinting	15
6.1	Positionsbestimmung	15
6.1.1	Nearest-Neighbor-Verfahren	15
6.1.2	Prohabilistisches-Verfahren	15
7	Fazit und Ausblick	17
	Bibliography	18

Abbildungsverzeichnis

1.1	Smartphoneabsatz in Deutschland	1
2.1	Außenhülle	5
2.2	Chipsatz mit Bluetooth-Modul	5
2.3	Ein iBeacon der Firma "estimote"	5
2.4	Aufbau des estimote-Beacons	5
3.1	Auswahlbildschirm der verschiedenen Templates	7
3.2	Beispiel eines Storyboards für iPhones	8
3.3	View Controller mit Constraints	9

1 Einleitung

1.1 Motivation

Die GPS-Navigation ist seit Jahren aus keinem Auto mehr wegzudenken. Wo früher Karten genutzt wurden und nach Straßennamen geschaut wurde, wird heute die Zieladresse in das Navigationssystem eingegeben und das System bestimmt selbstständig die aktuelle Position, die Zielposition und errechnet die bestmögliche Route. Ein Problem der GPS-Navigation ist jedoch, dass diese nur unter freiem Himmel akzeptabel funktioniert. In der Realität verbringen wir jedoch den Großteil unserer Zeit in Gebäuden, wo uns dieser Ansatz wenig weiterhilft.

Daher wäre es sinnvoll, eine Alternative zu GPS zu schaffen, welche diese Funktionen in Innenräume realisiert. Da man jedoch für Innenräume kein eigenes Navigationssystem kaufen möchte, liegt die Idee nahe, dieses Konzept auf einem Gerät zu realisieren, das sowieso schon viele Leute besitzen und auch schon für die GPS-Navigation nutzen. Das Smartphone. Wie in Abbildung 1 zu sehen, hat die Verbreitung der Smartphones in den letzten Jahren sehr stark zugenommen, sodass man annehmen kann, dass ein Großteil der potentiellen Nutzer der Indoor Positionierung auch ein Smartphone besitzt.

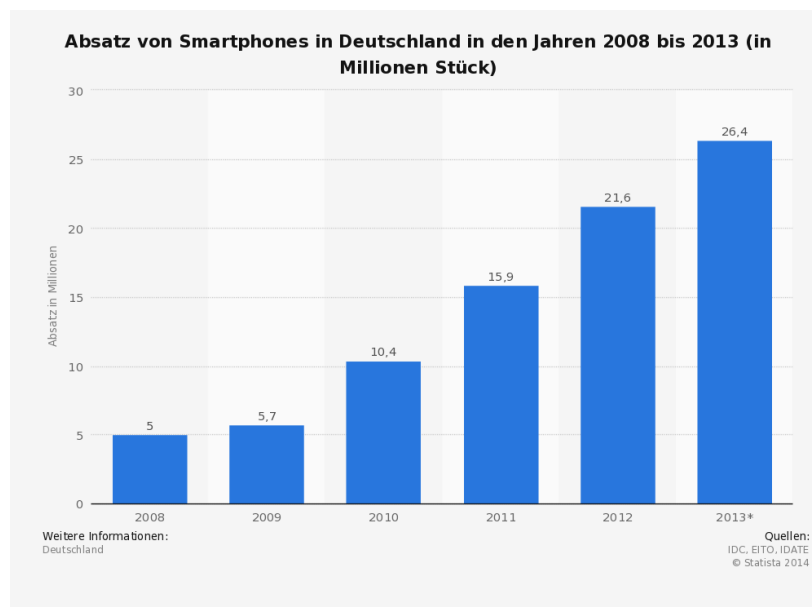


Abbildung 1.1: Smartphoneabsatz in Deutschland

Für die Indoor Positionierung würden verschiedene Technologien in Frage kommen, wie zum Beispiel Wireless LAN, RFID oder Bluetooth. Diese Technologien bieten sich an, da sie schon von Haus aus in vielen Smartphones integriert sind und so kein Bedarf an neuen Geräten oder Erweiterungen besteht.

Schlussendlich viel die Entscheidung der zu verwendenden Technologie auf Bluetooth, da dieses die höchste Verbreitung bietet und auch weitere Vorteile mit sich bringt. Zum einen ermöglicht Bluetooth eine schnelle und einfache Einrichtung und zum anderen benötigen die Bluetooth-Sendestationen nicht zwingend einen Stromanschluss, sondern erlauben auch einen Batteriebetrieb über mehrere Monate bis Jahre.

Die Positionierung in Innenräumen mittels Bluetooth ist ein relativ neuer Ansatz, welcher jedoch seit der Präsentation von Bluetooth Low Energy und der Vorstellung der iBeacons-Technologie von Apple immer mehr an Aufmerksamkeit gewonnen hat.

1.2 Ziele der Bachelorarbeit

Das Hauptziel dieser Arbeit ist es zu untersuchen, in wie weit sich Bluetooth Low Energy beziehungsweise die darauf basierende iBeacons-Technologie für eine akzeptable Indoor Positionierung eignet, um Endgeräte zum Beispiel in Verkaufsräumen zu orten und zu identifizieren.

Dabei soll untersucht werden, welches Verfahren sich dafür am Besten eignet und ob es Unterschiede zwischen verschiedenen Sende- und auch Empfangsgeräten gibt. Für die initialen Tests werden ausschließlich Apple-Geräte genutzt, da hier eine übersichtlichere Auswahl auf dem Markt ist, sodass man sich nicht mit unzähligen verschiedenen Messungen auseinander setzen muss.

Im Laufe der Bachelorarbeit soll deshalb eine iOS-Applikation entwickelt werden, welche eine Positionierung in einem Innenraum implementiert. Dabei wird die von Apple bereitgestellte CoreLocation-API genutzt, welche die Verarbeitung der iBeacon-Daten übernimmt. Die genutzten iBeacon-Sender kommen von Drittherstellern und sind derzeit noch in einem Vorserienstadium.

Zum Abschluss soll eine grundlegende Positionierung, eine Anzeige der aktuellen Position auf eine Karte und das Auslösen bestimmter Aktionen an festgelegten Orten implementiert sein.

2 Technologien

2.1 Bluetooth 4.0

Die Bluetooth-Version 4.0, oder auch Bluetooth Smart genannt, wurde 2009 final spezifiziert und wird seit Ende 2010 in Endgeräten eingesetzt. Dieser Standard beinhaltet neben dem klassischen Bluetooth, eine neue Version, mit dem Namen Bluetooth Low Energy, welche, wie der Name schon andeutet, einen sehr viel geringeren Stromverbrauch vorweist. Dabei ist der Stromverbrauch zwischen zwei und 100 mal geringer als beim klassischen Bluetooth.

2.2 Bluetooth Low Energy

Bluetooth Low Energy wurde Anfangs von Nokia unter dem Namen "Wibree" entwickelt. Die Zielsetzung dabei war es eine Technologie zu entwickeln, mit der sich Computer und Mobilgeräte schnell und einfach mit Peripherie-Geräten verbinden lassen. Das Hauptaugenmerk galt dabei dem geringen Stromverbrauch, kompakter Bauweise und den Kosten für die benötigte Hardware. Im Jahr 2007 wurden diese Spezifikationen dann in den, sich in der Entwicklung befindenden, Bluetooth-Standard 4.0 aufgenommen und daraufhin in Bluetooth Low Energy, oder kurz BLE umbenannt.

Bluetooth Low Energy arbeitet wie das klassische Bluetooth im 2,4 GHz Band, bringt aber in der Funktionsweise einige Unterschiede mit sich.

So wurde, im Vergleich zum klassischen Bluetooth, die Datenrate von bis zu 3 Mbit/s auf maximal 1 Mbit/s reduziert. Dies führt dazu, dass BLE zum Beispiel nicht für Headsets genutzt werden kann, da die zur Verfügung stehende Übertragungsrate nicht für die Audioübertragung ausreicht.

Die Vorteile die BLE mit sich bringt, liegen vor allem in der niedrigen Latenz, welche von 100ms auf bis zu unter 3ms reduziert wurde, und, wie bereits erwähnt, der Energieverbrauch drastisch gesenkt wurde.

Bluetooth Low Energy bietet darüber hinaus eine Vielzahl sogenannter GATT-Profile (Generic Attribute Profile). Die bereitgestellten GATT-Profile sind Richtlinien für die Bluetooth-Funktionalität, sprich, welche Daten übertragen werden und in welcher Form. Dies erlaubt eine einfache und schnell Interoperabilität zwischen verschiedenen Geräten. Ein Beispiel für ein GATT-Profil wäre zum Beispiel das "Heart Rate Profile", welches beispielsweise die Verbindung und Kommunikation eines Pulsmessgurtes mit einem Endgerät beschreibt. So wird sichergestellt, dass dieser Gurt mit jedem Endgerät auf die selbe Weise funktioniert.

2.2.1 iBeacons

Die iBeacons-Technologie wurde am 10. Juni 2014 von Apple auf der Worldwide Developers Conference vorgestellt. Diese basiert auf Bluetooth Low Energy und arbeitet mit einem von Apple entwickelten GATT-Profil.

Beacon bedeutet übersetzt "Leuchtfeuer" und die Funktionsweise der Beacons ist dem sehr ähnlich. Einmal in Betrieb genommen, sendet das Beacon kontinuierlich ein Signal, in welchem sich Daten zur Identifizierung des Beacons befinden.

Neben den Identifikationsdaten kann das Empfangsgerät noch weitere Größen bestimmen. Es ist so zum Beispiel möglich die ungefähre Entfernung einzuschätzen. In der iBeacons-API sind dafür vier verschiedene Zustände definiert: *Far*, *Near*, *Immediate* und *Unknown*. Diese Werte erlauben eine grobe Entfernungseinschätzung und für eine genauere Bestimmung lässt sich noch eine weitere Kenngröße bestimmen, der *Accuracy*-Wert. Dabei handelt es sich um eine ungefähre Entfernungsangabe in Metern, welche jedoch ausdrücklich nur zur Differenzierung zwischen zwei Beacons genutzt werden soll und keinesfalls eine genaue Entfernung angibt.

Daten	Format	Beschreibung	Beispiel
UUID	16-stellige Hexadezimalzahl	Identifizierung	3F4
Major	Integerzahl	Identifizierung eine Region	12
Minor	Integerzahl	Identifizierung eines einzelnen Beacons	132
Proximity	Drei Entfernungsstufen	Ungefähre Entfernung	Far, Near, Immediate und Unknown
Accuracy	Wert in Meter	Bestimmung der ungefähren Entfernung	1.243 m
RSSI	Signalstärke in dBm	Signalstärke des empfangenen Signals	-42 dBm

Die von dem Beacon gesendeten Daten lassen sich mit jedem BLE-kompatiblen Gerät empfangen.

Die großen Vorteile der iBeacons sind zum einen ihr kleiner Formfaktor, welcher es erlaubt die Beacons an fast jedem beliebigem Ort anzubringen, als auch ihr geringer Stromverbrauch, der es möglich macht, die Beacons bis zu mehreren Jahren mit einer Knopfzellenbatterie zu betreiben. Der Aufbau eines solchen Beacons lässt sich in Abbildung 2.3 sehr gut erkennen. Den Großteil des Beacons nimmt dabei die Batterie ein.

Unter genauerer Betrachtung des Chipsatzes in Abbildung 2.4, erkennt man, dass er im Grunde aus zwei Teilen besteht. Dem Bluetooth-Chipsatz, welcher an sich ist nur wenige Zentimeter groß und der Antenne, welche im vorderen Bereich der Platine eingearbeitet ist und die über welche letztendlich die Daten gesendet werden.



Abbildung 2.1: Außenhülle

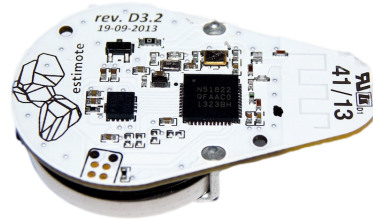


Abbildung 2.2: Chipsatz mit Bluetooth-Modul

Abbildung 2.3: Ein iBeacon der Firma "estimote"

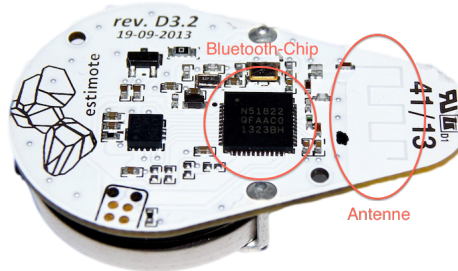


Abbildung 2.4: Aufbau des estimote-Beacons

2.3 iOS, OS X und Xcode

Für die Entwicklung der Applikation zur Indoor Positionierung war eine der Vorgaben, dass diese für iOS programmiert werden soll. Daher waren drei Dinge zwingend notwendig: ein Mac, Xcode und ein iOS-Gerät.

Für die Entwicklung setzte ich deshalb auf ein Macbook Pro mit installiertem Xcode und als iOS-Gerät setzte ich ein iPhone 5 ein. Als minimale iOS-Version musste iOS 7 verwendet werden, da die iBeacon-Features des CoreLocation-Frameworks (mehr dazu im Kapitel 2.4) erst ab dieser Version zur Verfügung stehen.

2.4 CoreLocation-Framework

2.4.1 iBeacons-API

2.4.2 Weitere API's

2.5 CoreData-Framework

3 Werkzeuge

3.1 Xcode

Xcode ist eine integrierte Entwicklungsumgebung von Apple, welche es ermöglicht iOS und OS X Applikationen zu programmieren, zu testen und zu debuggen. Standardmäßig werden dabei die Programmiersprachen *Objective C*, *C* und *C++* unterstützt.

Xcode stellt viele Features für die Programmierung bereit, wie zum Beispiel *code completion*, vorgefertigte *Templates*, einen umfangreichen *Debugger* und eine *iOS-Simulator* für das Testen der Applikationen.

Bei Erstellung einer neuen Applikation kann man unter mehreren Templates wählen, welche jeweils verschiedene Funktionen mit sich bringen. In Abbildung 3.1 lassen sich die verschiedenen Auswahlmöglichkeiten erkennen.

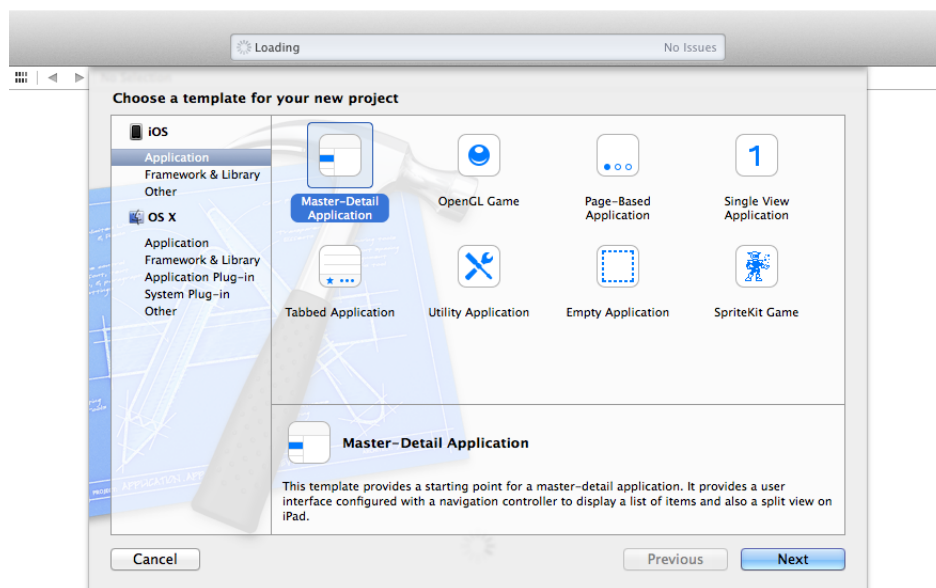


Abbildung 3.1: Auswahlbildschirm der verschiedenen Templates

Nach dem man das passende Template gewählt hat, werden die benötigten Dateien angelegt. Dazu gehören beispielsweise die *AppDelegate* und das *Storyboard*.

Die *AppDelegate*-Klasse steuert applikationsweite Ereignisse, wie etwa das Aufrufen und Schließen der Applikation. Außerdem wird durch die *AppDelegate* der aktuelle Zustand der Applikation gespeichert und wiederhergestellt.

Das Storyboard ist eine grafische Oberfläche für die Erstellung des User Interfaces. Es ermöglicht verschiedene Elemente wie zum Beispiel Views, Textfelder, Buttons oder Tabellen einzufügen und diese zu verbinden. Wie in Abbildung 3.2 zu erkennen, besteht das Storyboard aus mehreren View Controllern, die jeweils eine gezeigte Szene auf dem Gerät repräsentieren. Die einzelnen View Controller sind mit sogenannten *Segue's* verbunden, welche sich durch bestimmte Aktionen, wie zum Beispiel den Druck auf einen Button, auslösen lassen.

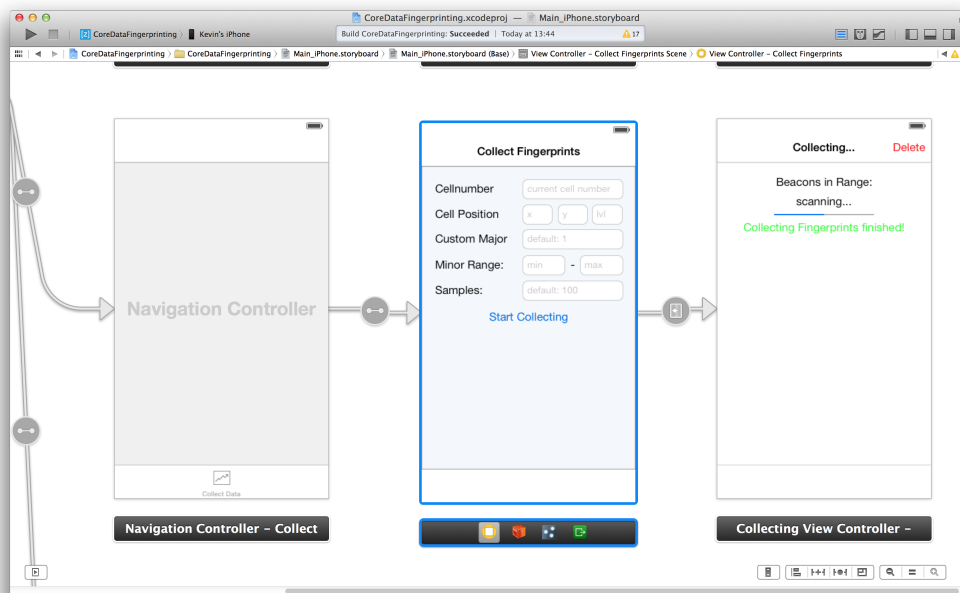


Abbildung 3.2: Beispiel eines Storyboards für iPhones

Das Storyboard bietet außerdem noch die Funktion des *Auto Layout*. Dabei werden sogenannte *Constraints* genutzt, welche die Positionsbeziehungen zwischen den einzelnen Elementen festlegen.

Aufgabe
r Constraints be-
schreiben

3.2 Objective C

3.3 Versionsverwaltung mit Git

3.4 iOS Developer Program

3.5 iPhone

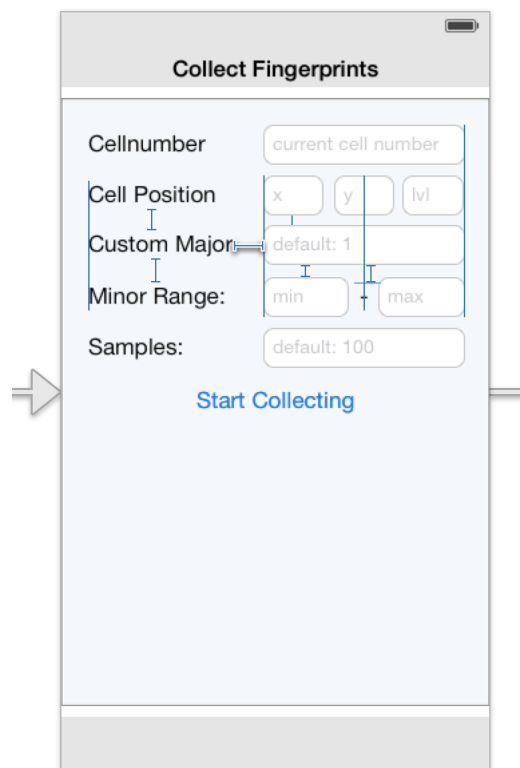


Abbildung 3.3: View Controller mit Constraints

4 Daten und Messungen

4.1 Mobile iBeacons

4.2 Stationäre iBeacons

4.3 Außenmessungen

4.4 Innenraummessungen

5 Umsetzung und Implementation

5.1 Ansatz zur Positionsbestimmung

5.2 Trilateration

5.3 Fingerprinting

6 Fingerprinting

6.1 Positionsbestimmung

6.1.1 Nearest-Neighbor-Verfahren

6.1.2 Prohabilistisches-Verfahren

7 Fazit und Ausblick

Erklärung

Ich versichere, dass ich die eingereichte Bachelor-Arbeit selbstständig und ohne unerlaubte Hilfe verfasst habe. Anderer als der von mir angegebenen Hilfsmittel und Schriften habe ich mich nicht bedient. Alle wörtlich oder sinngemäß den Schriften anderer Autoren entnommenen Stellen habe ich kenntlich gemacht.

Osnabrück, den 24.09.2013

(Kevin Seidel)