**Requirements Specification**

Semesterprojekt SS 2018

Brain2Machine Interface

Version 1

# Inhaltsverzeichnis

Inhaltsverzeichnis 2

1 Einleitung 3

1.1 Gültigkeit des Dokuments 3

1.2 Begriffsbestimmungen und Abkürzungen 3

1.3 Zusammenhang mit anderen Dokumenten 3

1.4 Überblick über das Dokument 3

2 Allgemeine Beschreibung des Produkts 4

2.1 Projektumgebung 4

2.1.1 Übersicht 4

2.1.2 Kontaktdaten 5

2.2 Zweck des Produkts 5

2.3 Überblick über die geforderte Funktionalität 5

2.4 Allgemeine Einschränkungen 5

2.5 Vorgaben zu Hardware und Software 6

2.6 Benutzer des Produkts 6

3 Detaillierte Beschreibung der geforderten Produktmerkmale 7

3.1 Lieferumfang 7

3.2 Erstentwürfe 7

3.3 Abläufe (Szenarien) von Interaktionen mit der Umgebung 8

3.4 Externe Schnittstellen des Produkts 9

3.4.1 Benutzerschnittstellen (User Interfaces) 9

3.4.2 Systemschnittstellen 11

4 Vorgaben an die Projektabwicklung 12

4.1 Anforderungen an die Realisierung 12

4.1.1 Hardware 12

4.1.2 Software 12

4.2 Fertige und zugekaufte Komponenten 12

4.3 Lieferbedingungen 12

5 Verpflichtungen des Auftraggebers 13

# Einleitung

## Gültigkeit des Dokuments

Dieses Pflichtenheft ist für das gesamte Projekt Brain2Machine Interface (Version 1) gültig. Es setzt auf kein bestehendes Pflichtenheft auf. Für Änderungen ist das Projektteam (Paul Schmutz, Boris Fuchs) zuständig.

## Begriffsbestimmungen und Abkürzungen

|  |  |
| --- | --- |
| Interface | Schnittstelle zwischen verschiedenen Anwendungen bzw. Hard- und Software |
| EEG | Elektroenzephalografie; ist eine Methode zur Messung der elektrischen Aktivität des Gehirns durch Aufzeichnung der Spannungsschwankungen mithilfe von Elektroden, die am Kopf angebracht werden |
| OpenEEG | Ein Open Source Projekt zur Entwicklung kostengünstiger EEG Geräten und dazugehöriger kostenloser Software  <http://openeeg.sourceforge.net/> |
|  |  |
|  |  |
|  |  |

## Zusammenhang mit anderen Dokumenten

Derzeit liegt weder ein Angebot noch ein Lastenheft vor.

## Überblick über das Dokument

* Allgemeine Beschreibung des Produkts
* Detaillierte Beschreibung der geforderten Produktmerkmale
* Vorgaben an die Projektabwicklung (Hardware, Software, Betriebssystem)
* Verpflichtungen des Auftraggebers

# Allgemeine Beschreibung des Produkts

## Projektumgebung

### Übersicht



**FH Hagenberg**

Auftraggeber, Projektkoordination & -betreuung

Dipl. Ing. Dr. Erik Sonnleitner



Auftragnehmer

Boris Fuchs

Paul Schmutz

Projektauftrag,

Projektkoordination,

Beratung,

Fachwissen

### Kontaktdaten

|  |
| --- |
| **Dipl. Ing. Dr. Erik Sonnleitner**  Telefon: +43 5 0804 22823  E-Mail: erik.sonnleitner@fh-hagenberg.at |
| **Boris Fuchs**  Telefon: +43 676 884 003 638  E-Mail: boris.fuchs@students.fh-hagenberg.at |
| **Paul Schmutz**  Telefon: +43 680 303 06 76  E-Mail: paul.schmutz@students.fh-hagenberg.at |

## Zweck des Produkts

Der Zweck des Produkts liegt darin, aufzuzeigen,

* in welchem Umfang
* in welcher Form
* und mit welcher Genauigkeit

Daten aus dem Open Source Projekt Olimex EEG ausgelesen werden können.

Die Kernaufgabe besteht darin, die Daten entsprechend aus dem Olimex EEG extrahieren zu können. Weiters müssen diese Daten verarbeitet werden (Fouriertransformation, Filter, Korrelation, Schwellwerte, … ), damit unterschiedliche Gesten bzw. auch Gedanken des EEG Benutzers erkannt werden können. Die Unterscheidung verschiedener am Kopf gemessenen Zustände soll anschließend dazu dienen, eine Demoapplikation anzusteuern, um die Auswirkungen von vom Benutzer bewusst erzeugten Gedanken oder Gesten zu visualisieren.

## Überblick über die geforderte Funktionalität

Die Kernfunktionalität des Produkts besteht darin, durch messbare Gedanken oder Gesten von Personen eine Demoapplikation anzusteuern.

Im Genaueren ist es dazu erforderlich, folgende Teilfunktionalitäten zu erfüllen

* Empfangen von Daten über den USB-Port, an welchem das Olimex EEG angeschlossen ist, in Form eines Byte-Stroms
* Umwandeln des Byte-Stroms in Datenpakete in Form von Objekten einer höheren Programmiersprache
* Analyse der Daten mithilfe von Fouriertransformation, Korrelation, Filterung und ähnliche Werkzeuge der Signalverarbeitung

Vor der Verwendung von höheren Programmiersprachen dient zuerst Matlab als hilfreiches Werkzeug, um die empfangenen EEG-Daten zu visualisieren und passende Kriterien zur Detektion bestimmter Gesten und Gedanken festzulegen.

## Allgemeine Einschränkungen

Welche Demoapplikation konkret entwickelt wird, steht dem Projektteam frei. Die Kernaufgabe der Demoapplikation sowie die Technologie kann vom Projektteam bestimmt werden.

Essenziell ist jedoch die Erfüllung der geforderten Funktionalität (siehe 2.3) und dass die Demoapplikation in ausreichendem Maße die Funktionsfähigkeit des EEGs repräsentiert, damit diese auch von Personen ohne Bezug auf die EEG-Technologie bedient und verstanden werden kann.

## Vorgaben zu Hardware und Software

Die Voraussetzung für den Entwicklungsrechner ist in diesem Projekt Windows, da auch die Olimex-Benutzeranleitung die Angaben für Windows spezifiziert. Jedoch stehen Open EEG Projekte auch für andere Betriebssysteme zur Verfügung. Eine Auflistung von bestehenden Projekten ist unter <http://openeeg.sourceforge.net/doc/sw/> abrufbar.

Zur Einführung in die Verarbeitung des Olimex EEG Signals, stehen Softwarehilfsmittel zur Verfügung:

* Matlab:  
  Matlab ist ein hilfreiches Tool, um die Rohsignale aus dem EEG auszulesen und diese Daten zu visualisieren. Vorteil damit ist, dass bereits bestimmte Eigenschaften des Signals betrachtet werden können und Analysen (Fourier, Filterung) durchgeführt werden können.
* ElectricGuru:  
  Die Software (aufgelistet auf der Open EEG Webseite) ist hilfreich, um den Zeitverlauf des EEGs in Echtzeit zu verfolgen. Damit können die Auswirkungen auf verschiedene Gesten sofort visuell wahrgenommen und dokumentiert werden.

Die Wahl der Technologie für die zu entwickelnde Demoapplikation steht frei. Das Projektteam sieht folgende zu erstellende Software vor:

* Demoapplikation in Java
  + Visualisierung von Gesten mit Text/grafischen Elementen
  + Zum Testen und Optimieren der Gestenerkennung
* Steuerung eines Lego Mindstorms Kran (Java)
  + Einsatz der funktionsfähigen Gestenerkennung
  + Steuerung des Krans durch Gesten

Der Grundbaustein der Hardware ist das Open EEG kompatible EEG Gerät von Olimex (<https://www.olimex.com/Products/EEG/OpenEEG/EEG-SMT/open-source-hardware>).

Um Vergleiche festzustellen und Optimierungen an den Elektrodenpositionen vorzunehmen, stellt die FH Hagenberg weiters ein EEG von Emotiv Insight bereit.

Genaueres zum Lego Mindstorms befindet sich unter <https://www.lego.com/en-us/mindstorms>.

## Benutzer des Produkts

Da das Projekt durch die FH (Prof. Sonnleitner) in Auftrag gegeben wurde, steht das Produkt Angestellten sowie Studierenden der FH Hagenberg zur Verfügung, denen in späterer Folge die gewonnenen Erkenntnisse aus dem Bereich EEG bzw. OpenEEG sowie dessen Vor- und Nachteilen zur Verfügung gestellt werden können.

# Detaillierte Beschreibung der geforderten Produktmerkmale

## Lieferumfang

Das fertige Endprodukt enthält zwei wesentliche Bestandteile:

* Eine ausführbare Java-Datei
* Benutzerhandbuch

## EEG & Elektrodenpositionierung

### EEG (Olimex)

Im gesamten Aufbau sieht das EEG von Olimex folgendermaßen aus:



Die Stromversorgung und Datenübertragung vom EEG Gerät wird mittels einer USB Verbindung erzielt. Die Elektroden dienen zur Spannungsmessung am Kopf des Benutzers.



4x Aktive Elektroden 🡪

1x Passive Elektrode

USB-B

**Anmerkung**: In diesem Projekt werden 4x aktive und 1x passive Elektroden verwendet. Laut Olimex sind auch 5x passive Elektroden zum Betreiben der Messungen zulässig.

### Elektrodenpositionierung

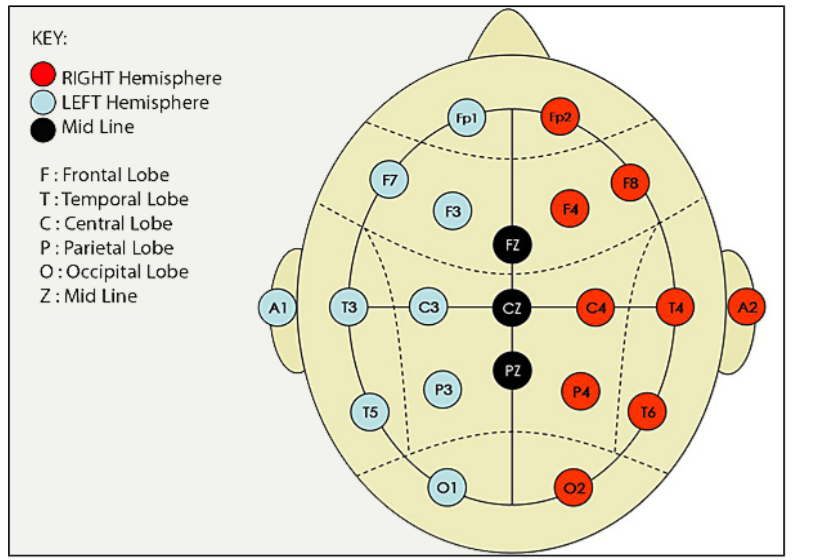
Der Signalverlauf der erhaltenen EEG Daten hängt stark von der Positionierung der Elektroden ab.

Folgende Anhaltspunkte sind gegeben:



Im Rahmen der Projektdurchführung werden unterschiedliche Konstellationen getestet und analysiert.

Hauptsächlich kommen jedenfalls die vorderen Kopfbereiche zum Einsatz, da hier die Signalverläufe von Gesten wie zB Augenbewegungen sehr gute Eigenschaften (hohe Ausschläge, deutlich unterscheidbar) zur Detektion liefern.



**CH2+**

**CH1+**

**CH1-**

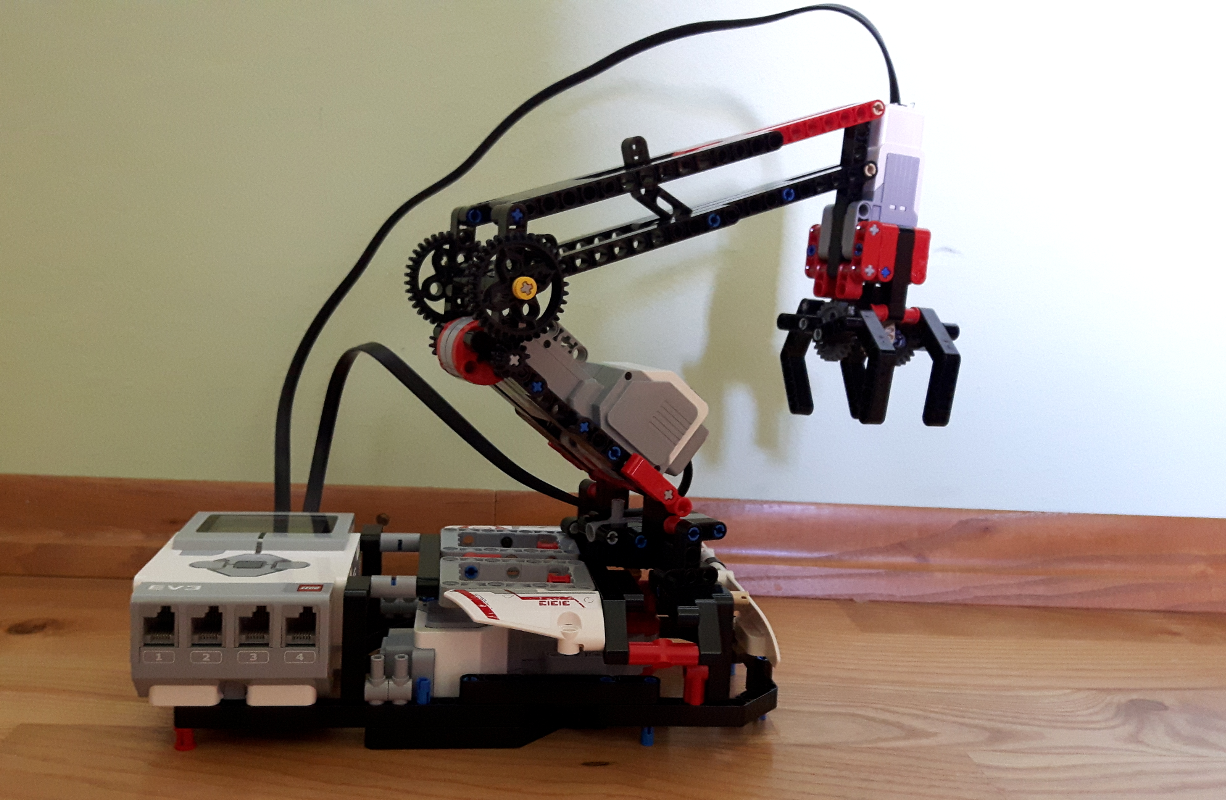
**CH2-**

Um die Elektroden an der gewünschten Position zu befestigen, werden diese unter einer eng anliegenden Schwimmhaube ausgerichtet.



## Abläufe (Szenarien) von Interaktionen mit der Umgebung

Der vom Benutzer zu steuernde Lego Mindstorms Kran sieht folgendermaßen aus:



Dem Benutzer des Systems stehen bestimmte Funktionalitäten zur Verfügung, welche für folgende Szenarien geeignet sind:

Lego Kransteuerung



Benutzer  
(mit EEG Elektroden)

<<include>>

<<include>>

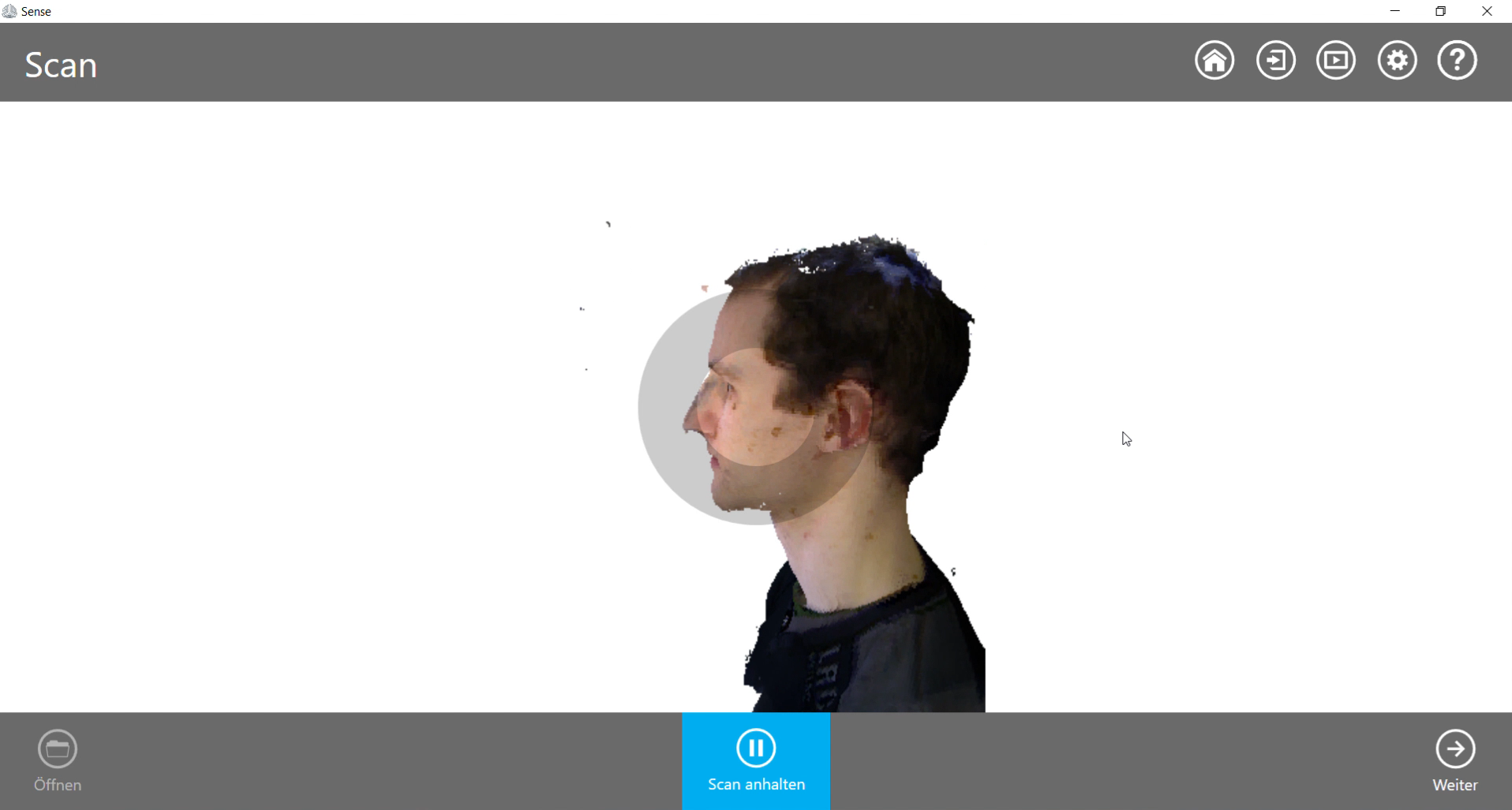
## Externe Schnittstellen des Produkts

### Benutzerschnittstellen (User Interfaces)

Da das Projekt überwiegend Hardware-Charakter aufweist, fallen kaum Entwürfe von Benutzerschnittstellen an.

#### Sense Software

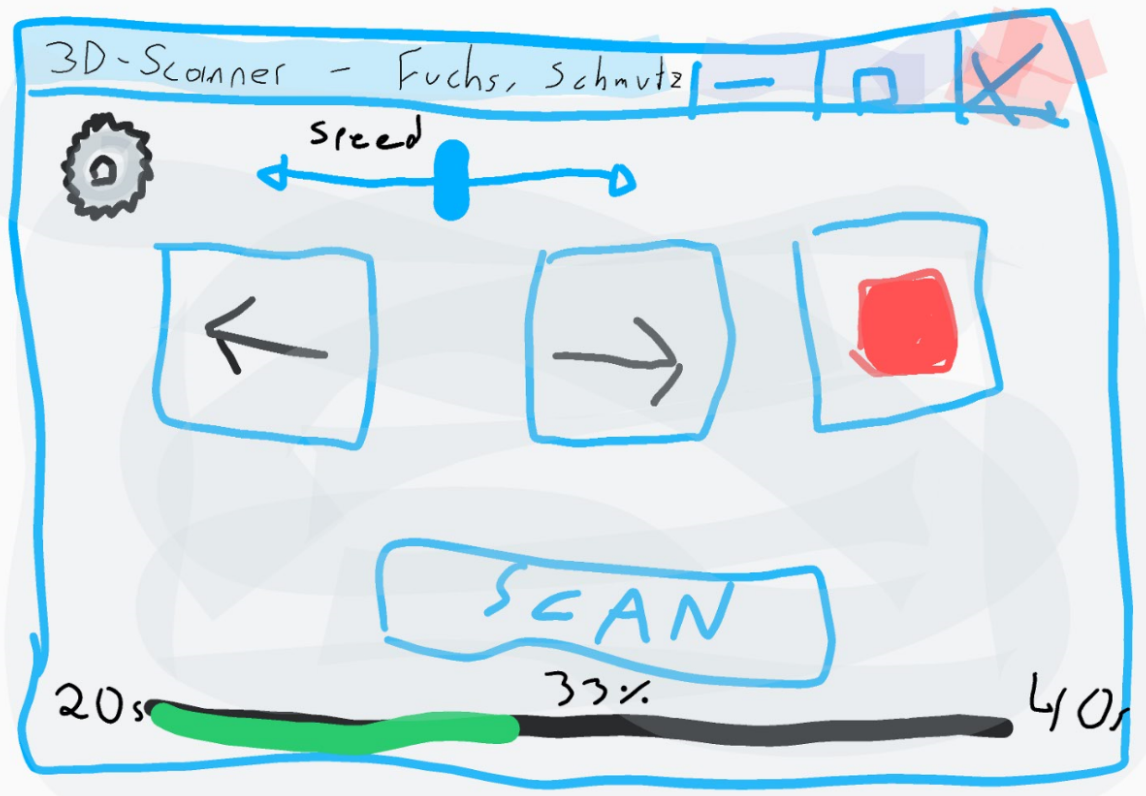
Die mit dem Sense 3D-Scanner ausgelieferte Software erfasst die vom Scanner aufgefassten Daten und verarbeitet sie zu einem 3D Modell. Die Benutzeroberfläche ist dabei recht übersichtlich aufgebaut:



Button zum Start/Stopp/Fortsetzen eines Scan-Vorgangs

#### 3D-Scanner-Programm

Da für den Aufbau sowohl die Sense Software sowie die Lego Motoren angesprochen werden müssen, vereint ein vom Projektteam zu entwerfendes Programm diese Komponenten.



**1. Einstellungen**: Zur Abänderung von derzeit noch nicht näher definierten Parametern, die den Scan-Vorgang beeinflussen (sollten zu wenige Parameter gefunden werden, entfällt dieses Menü)

**2. Geschwindigkeit**: Dient zur Anpassung der Geschwindigkeit der Motoren für das manuelle Bewegen der Plattform (siehe 3.).

**3. Manuelle Bewegung**: Treibt die Motoren der Plattform an, damit diese gegen (linker Button) bzw. in den Uhrzeigersinn (rechter Button) mit der eingestellten Geschwindigkeit (siehe 2.) angetrieben wird.

**4. Stopp**: Stoppt die Motoren sowie einen möglicherweise aktiven Scan-Vorgang (auch den Scan-Vorgang der Sense Software).

**5. Auto-Scan**: Startet den Scan-Vorgang der Sense Software und bewegt die Plattform einmal rund um die Person/das Objekt. Nach Rundumerfassung wird der Scan-Vorgang der Sense Software ebenfalls gestoppt.

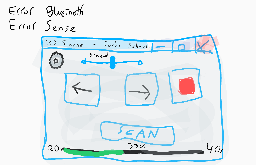
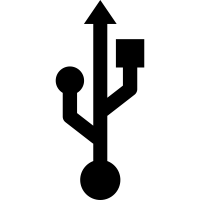
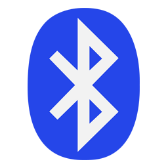
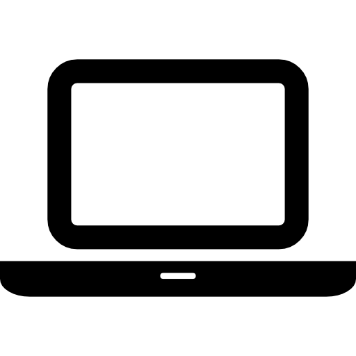
**6. Scan-Fortschritt**: Zeigt an, wie lange der Auto-Scan aktiv ist bzw. noch dauern wird.

### Systemschnittstellen

Die in der zu entwickelnden Anwendung enthaltenen Schnittstellen sind wie folgt strukturiert:

EV3 Brick

PC



Sense Scanner

**Anmerkungen**

Der Sense 3D-Scanner wird über USB mit dem PC verbunden. Diese Verbindung ist notwendig zur Stromversorgung sowie Datenübertragung des Scanners an die Sense Software.

Gleichzeitig wird der PC über Bluetooth mit dem Lego EV3 Brick verbunden. Auf dem Brick läuft Betriebssystem leJOS, wodurch über Java RMI die am Brick angeschlossenen Motoren vom PC aus angesteuert werden können.

# Vorgaben an die Projektabwicklung

Im Folgenden werden die Voraussetzungen zur Abwicklung des Projekts hinsichtlich technischer sowie organisatorischer Bedingungen dargestellt.

## Anforderungen an die Realisierung

RXTX for Java

Matlab mit Psychotoolbox 3

ElectricGuru

Olimex EEG

Freie Erstellung von Software für Olimex EEG

Emotiv Insight zum Testen + gebunden an dessen Software

### Hardware

Zusammengefasst bestehen die Hardware-Anforderungen aus

* Entwicklungsrechner (Notebook)
* Sense 3D-Scanner
* Lego Mindstorms Set

### Software

#### Betriebssystem

Für das Betriebssystem des Entwicklungsrechners wird bevorzugt **Windows 10** verwendet. Grund dafür ist, dass auch die Sense Software nur für Windows geeignet ist.

Bezüglich Betriebssystem stimmen die Anforderungen an das **Endsystem** mit jenen des Entwicklungssystems überein.

## Fertige und zugekaufte Komponenten

Ein Zukauf von Hardware-Komponenten wird durch die FH Hagenberg abgewickelt.

Das Projektteam hat dazu Kostenschätzungen vorzunehmen und mit dem Auftraggeber die Bestellung konkreter Teile zu vereinbaren. Nach Möglichkeit sollen die Zukäufe über Internetbestellungen abgewickelt werden. Bei Bedarf kann das Projektteam nach Absprache mit dem Auftraggeber Komponenten eigenständig besorgen und finanziell durch Nachweis mit entsprechenden Belegen mit dem Auftraggeber abgleichen.

## Lieferbedingungen

Das Projekt wird spätestens am **09.02.2018** vollständig an den Auftraggeber übergeben.

Das Projekt gilt als abgenommen, wenn sich Projektdokumentation und Source-Code zum Stichtag auf dem angegebenen SVN-Repository befinden.

SVN-Repository: <https://svn01.fh-hagenberg.at/mcm/JG16W17P06>

# Verpflichtungen des Auftraggebers

Nachdem sowohl Auftragnehmer und Auftraggeber der FH Hagenberg angehören, verpflichtet sich der Auftraggeber die Auftragnehmer durch Vermittlung von Know-How und Bereitstellung von Hilfsmitteln zu unterstützen.

Für dieses Projekt stellt die FH Hagenberg ein Lego Mindstorms Set und den Sense 3D-Scanner mit dazugehöriger Software bereit. Weiters werden auch alle restlichen Bauteile für das Schienensystem durch die FH Hagenberg finanziert.

Der Auftraggeber kann bei Bedarf aufgrund folgender Aspekte vom Projektteam kontaktiert werden:

* Versorgung mit spezieller Software/Hardware zur Abwicklung des Projekts
* Organisatorische (nicht-technische) Beratung zu Fragen zur Realisierung des Projekts