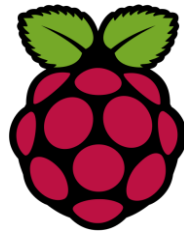


***„RPi-404“***

**Projektkonzept ITS/MOSY SoSe 2018**

**Projektgruppe C**

**Stand: 21.04.2018**



# Inhaltsverzeichnis

## Inhalt

Arbeitstitel.....	3
Gruppenmitglieder .....	3
Zeitraumen des Projekts .....	3
Projektziel.....	3
Anforderungsanalyse .....	3
Technische Rahmenbedingungen .....	4
Hardware.....	4
Software .....	4
Technisches Konzept .....	4
Steuerung des Ultraschall Abstandssensors für die Radarfunktion .....	5
Videoübertragung .....	5
Carry Me Home - Funktion.....	5
Ansteuerung der Skripte .....	5
Bedienkonzept.....	5
Meilensteine.....	5
Anhang .....	6

## Arbeitstitel

Der Arbeitstitel des Projekts lautet „**RPi-404**“.

Für die Abschlusspräsentation auf der Jahresausstellung am **10.07.2018** wird gegebenenfalls ein Titel gewählt, welcher den Besuchern der die Zielsetzung des Projekts näher erläutert.

## Gruppenmitglieder

Die Projektgruppe C besteht aus folgenden Mitgliedern:

Stefan Wyrembek	mail: stefan.wyrembek@haw-hamburg.de
Pieter Haase	mail: pieter.haase@haw-hamburg.de
Dario Schüler	mail: dar.schueler@gmail.com
Volker Thum	mail: volkerthum@web.de

## Zeitraahmen des Projekts

Das Projekt hat einen Zeitrahmen vom **17.04.2018** bis zu der Jahresausstellung am **10.07.2018**.

Die detaillierte Zeitplanung und die hiermit zusammenhängenden Meilensteine werden in dem Kapitel „[Zeitplan](#)“ beschrieben

## Projektziel

Besuchern der Jahresausstellung wird es ermöglicht, eine mobile Roboterplattform mit integrierter Kamera anhand einer „Mobile App“ zu steuern. Die von dem RPi-404 aufgenommenen Bilder der Umgebung, werden dem Benutzer live auf einem Smartphone bereitgestellt, womit Sichtkontakt mit der Plattform nicht nötig ist. Die Kameraposition ist zur besseren Orientierung steuerbar. Eine weitere Funktion ist das Radarsystem, welches einem Benutzer erlaubt, Objekte in der Dunkelheit zu erkennen und zu umfahren. Der in der App integrierte „Home-Button“ veranlasst den RPi-404 autonom zu einer vorher definierten „Dockingstation“ zu fahren.

## Anforderungsanalyse

Es wird eine „Mobile App“ für Smartphones entwickelt welche es ermöglicht die mobile Roboterplattform durch einen Raum zu bewegen. Die Applikation stellt dem Benutzer zusätzlich ein live-Video anhand der auf der Plattform installierten Kamera zur Verfügung. Zur besseren Orientierung kann der Benutzer durch Schwenkbewegungen des Smartphones die Kamera steuern. Der zweite Modus der Applikation beinhaltet ein Radarsystem, welches Hindernisse durch Ultraschall identifiziert und visuell darstellt. Somit soll es dem Benutzer ermöglicht werden, den RPi-404 auch durch eine lichtschwache Umgebung zu navigieren. Eine weitere Funktion

(„Carry Me Home“) soll den RPi-400 in einen autonomen Modus versetzen, in welchem er eine vorher definierte „Dockingstation“ erkennt und ansteuert. Alle Funktionen können vom Benutzer in der Applikation separat ausgewählt werden.

## Technische Rahmenbedingungen

### Hardware

Folgende Hardware wird zur Umsetzung des Projekts verwendet:

- Raspberry Pi 3 Model B
- Breadboard für die Elektronik
- Mobile Plattform
- H-Brücke zur Ansteuerung der Motoren
- Kamera
- Servomotor für die Kamera
- Ultraschall Abstandsensor für das Radarsystem
- Stromversorgung des RPi-404 in Form eines geeigneten Akkus
- Smartphone

### Software

Folgende Software wird zur Umsetzung des Projekts verwendet:

- Raspbian (Betriebssystem für den Raspberry Pi)
- On-Board W-Lan des Raspberry Pi
- Python 3 als Programmiersprache für die Sensorik und die Ansteuerung der Elektronik
- „OpenCV library“ für die Erkennung von Objekten durch die Kamera
- Android Entwicklungsumgebung (Adroid Studio)

## Technisches Konzept

Eine exemplarische Skizze der Schaltung ist dem Anhang (Abbildung 1) zu entnehmen.

### Betriebssystem

Für den Raspberry Pi 3 B wird „Raspbian“ als Betriebssystem gewählt.

### Steuerung der Motoren

Die Motoren für die Mobilität sowie für die Kamera werden durch Python 3-Skripte über die GPIO-Ports angesteuert. Die Spannungsversorgung der Motoren wird durch einen Akku sichergestellt.

## Steuerung des Ultraschall Abstandssensors für die Radarfunktion

Der Sensor wird über die 5V Spannungsversorgung des Raspberry Pi betrieben. Die von dem Sensor verarbeiteten Spannungswerte werden von einem GPIO-Pin eingelesen und durch ein Python-Skript verarbeitet.

## Videoübertragung

Das Videobild wird von einer USB-Kamera aufgezeichnet, welche mit dem Raspberry Pi verbunden ist. Das Bild wird über W-Lan auf ein Smartphone übertragen.

## Carry Me Home - Funktion

Die „Carry Me Home – Funktion“ soll den RPi-404 automatisch in eine vorher definierte Dockingstation fahren. Für diese Funktion muss die Dockingstation durch die Kamera erkannt werden um das Ziel daraufhin ansteuern zu können. Für die Erkennung der Station wird die Bibliothek „OpenCV“ in ein Python-Skript implementiert.

## Ansteuerung der Skripte

Die Ausführung aller Skripte wird über eine Android-Applikation gestartet. Der Benutzer hat in der Applikation die Möglichkeit zwischen den jeweiligen Funktionen zu wählen. Die Steuerbefehle werden über W-Lan an den Raspberry Pi gesendet.

## Bedienkonzept

Nach dem Start der Applikation wird dem Benutzer das von der Kamera aufgezeichnete Bild angezeigt. Auf das Bild wird ein virtueller Joystick überlagert, welchem dem Benutzer die Steuerung des RPi-404 ermöglicht. Durch schwenken des Smartphones rotiert die Kamera. Durch die Aktivierung des zweiten Modus wird anstatt des Kamerabilds ein Radar angezeigt, welches Objekte visualisiert, die durch den Ultraschallsensor erfasst wurden. Wenn die „Carry Me Home – Funktion“ aktiviert wird, ist eine manuelle Steuerung nicht mehr möglich und der RPi-404 fährt autonom, bis diese Funktion unterbrochen wird. In diesem Modus ist das Kamerabild sichtbar.

## Meilensteine

08.05. Übertragung des Live-Bildes vom Raspberry an das Smartphone

22.05. Steuerung der Kamera und der Motoren

05.06. Implementierung des Radars

12.06. Carry Me Home – Funktion

Alle folgenden Termine werden für das „Testing“ und das „Debugging“ veranschlagt.

Anhang

Schaltplan mit „Fritzing“

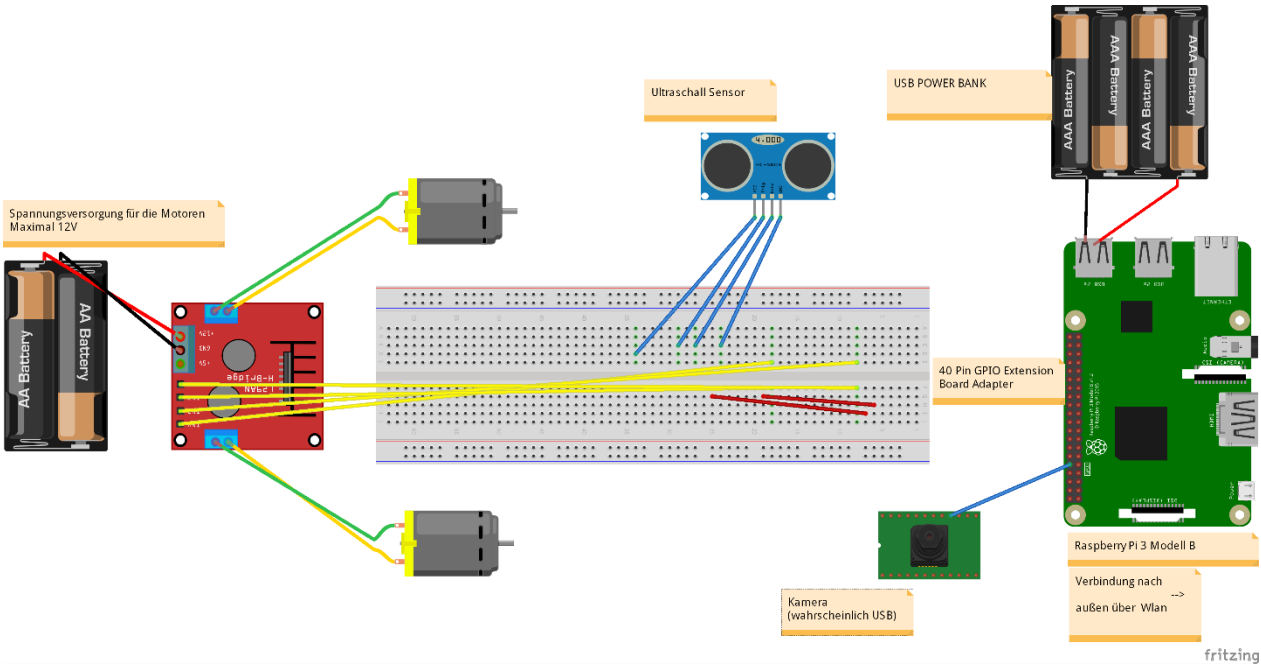


Abbildung 1