

PiSense mit Quadrocopter

Arbeitsbereich: WebProjectManagment von Dominik Heinrich

Abstract

Teammitglieder

Philipp Gackstatter — Jascha Petter — Marcel Frueh — Christoph Weik — Dominik Heinrich

Contents

Einleitung	1
1 Architektur des Projektes	1
2 Arbeitsbereich Embedded Systems	2
3 Arbeitsbereich Datenbanken	2
4 Arbeitsbereich Graphical User Interface	2
5 Arbeitsbereich Application	3
6 Arbeitsbereich WebProjectManagment	3

1. Architektur des Projektes

Einleitung

In dem Programmierprojekt PiSense mit Quadrocopter im Arbeitsbereich Eingebettete Systeme der Universität Tübingen gab es dieses Semester folgende Aufgabenbereiche:

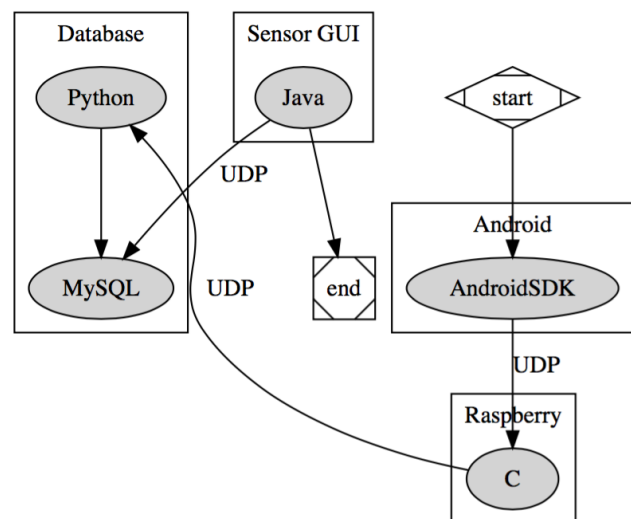
Embedded Systems Der Anpassung der Low Level Treiber des Quadrocopters in C, um Sensordaten in Echtzeit abrufen und auswerten zu können.

Datenbanken Der Entwicklung und Anbindung einer MySQL Datenbank zur Verwaltung der abgerufenen Sensordaten in Python.

Graphical User Interface Der Weiterentwicklung einer grafischen Oberfläche zur Darstellung der Sensordaten mit Hilfe von Java und JFreeChart.

Application Der Entwicklung und Anbindung einer mobilen Applikation zur Ansteuerung des Quadrocopters mit Android SDK.

WebProjectManagment Die Erstellung einer Webapplikation mit Hilfe von HTML, CSS, PHP, MySql LAMP sowie Engagement im Bereich der Organisation.



Programmierprojekt Quadrocopter

Hinweis: Dieses Bild wurde mit DOT GraphViz erstellt, siehe Figure 1.

Meine Veränderungen/Neuerungen:

Meine Aufgabe war es die Daten aus der Datenbank auszulesen und anzuzeigen. Zuerst kam die Implementierung der Konfigurationsdateien, also das erstellen, auslesen und importieren. Danach folgte die automatische Anpassung der Graphen. Diese lagen zuvor übereinander und konnten zu- und weggeschaltet werden. Um eine Überlagerung zu vermeiden, musste die gesamte Skala in die zugehörigen Skalen der aktuell zugeschalteten Graphen unterteilt werden. Schließlich folgte das Auslesen der Daten aus der Datenbank. Um eine Echtzeit-Anzeige von Daten zu ermöglichen die auch in Echtzeit in die Datenbank geschrieben werden, muss die Datenbank immer wieder ausgelesen werden um auf neue Daten überprüft zu werden. Da noch keine Dokumentation des Projekts vorhanden war, erstellte ich diese in Javadoc.

Neben diesen übergreifenden Aufgaben gab es auch kleine Änderungen, wie etwa Fehlerbehebungen, löschen von nicht mehr benötigten Funktionen, neuordnen von Code und Projektdateien, Änderungen an der GUI und verbessern der Usability. Für die GUI Programmierung möchte ich mich bei meinen Vorarbeitern Alexander Deitche und Juan-Carlos Barradas-Palmeros bedanken.

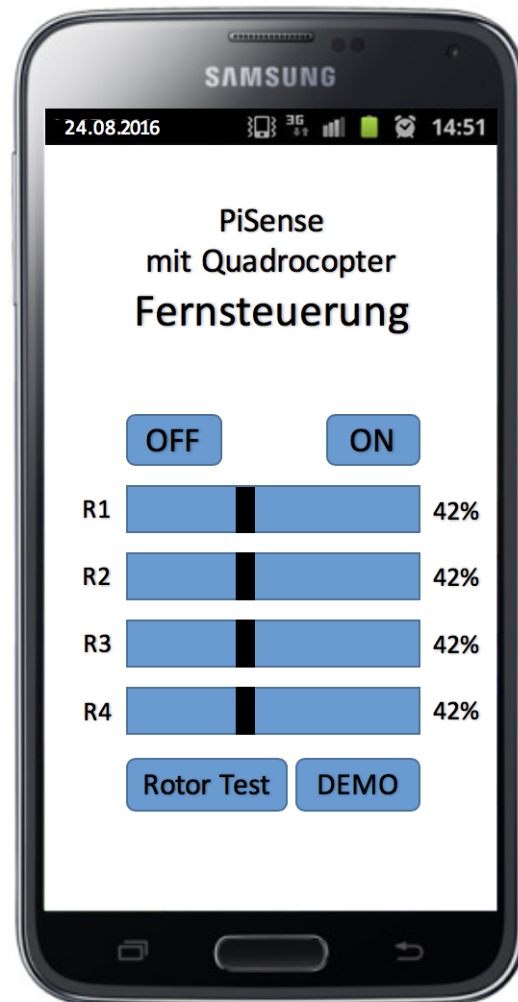
5. Arbeitsbereich Application

Bild: Applikation

6. Arbeitsbereich WebProjectManagment**Funktionsweise:**

Im Bereich WebProjectManagment wurde eine Webapplikation erstellt, in dem es verschiedene Funktionen gibt. Zum einen hat man die Möglichkeit sich zu registrieren und sich danach einzuloggen. Je nachdem welche Berechtigungen man dann hat, hat man Zugriff auf unterschiedliche Inhalte. Zum Beispiel ist das Ticket bearbeiten nur im "admin-modus" möglich.

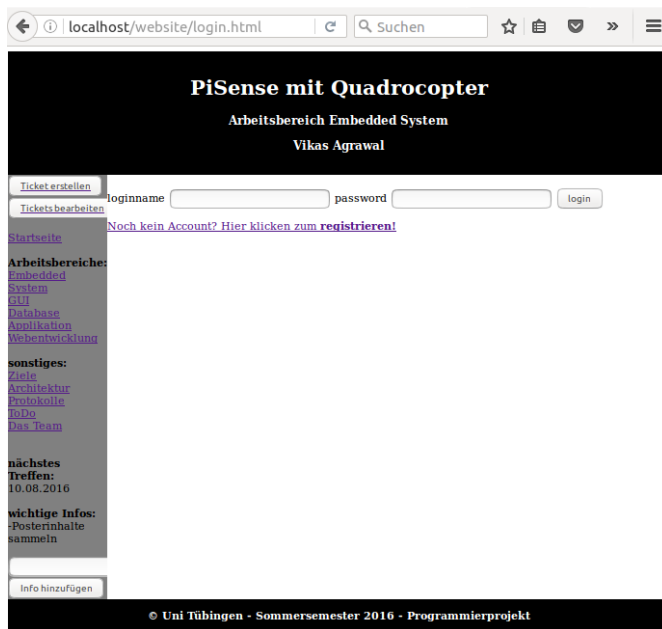


Bild: Screenshot login.html

Technische Grundlagen:

Die Webapplikation wurde auf einem Apache server gestartet mit Hilfe von LAMP(Linux Apache MySQL PHP). Mit LAMP wird auch die Datenbank verwaltet. Die Weboberfläche wurde mit HTML 5 und CSS gestaltet. Die Kommunikation mit MySQL läuft über die Sprichtsprache PHP. Um die Übersichtlichkeit der einzelnen Seiten zu optimieren haben wir alle Seiten in PHP geschrieben und am Anfang, sowie am Ende der jeweiligen Seiten ein `include("header.php")` und `include("footer.php")` eingefügt, was uns erlaubt die ganze Navigationszeile, das Ticketsystem, den Loginbereich sowie das Grunddesign auszulagern. Der Vorteil liegt darin dass man nur noch eine bzw. zwei Dateien ändern muss und nicht für eine kleine Änderung, alle Seiten welche die Bereiche benutzen. Im Prinzip der gleiche Vorteil welchen wir auch durch die CSS auslagerung haben. Möchten wir unser design ändern, können wir dies in der CSS datei machen und alle Seiten, welche ja die style.css eingefügt haben, führen dann die Änderung automatisch aus.

Die Loginfunktion:

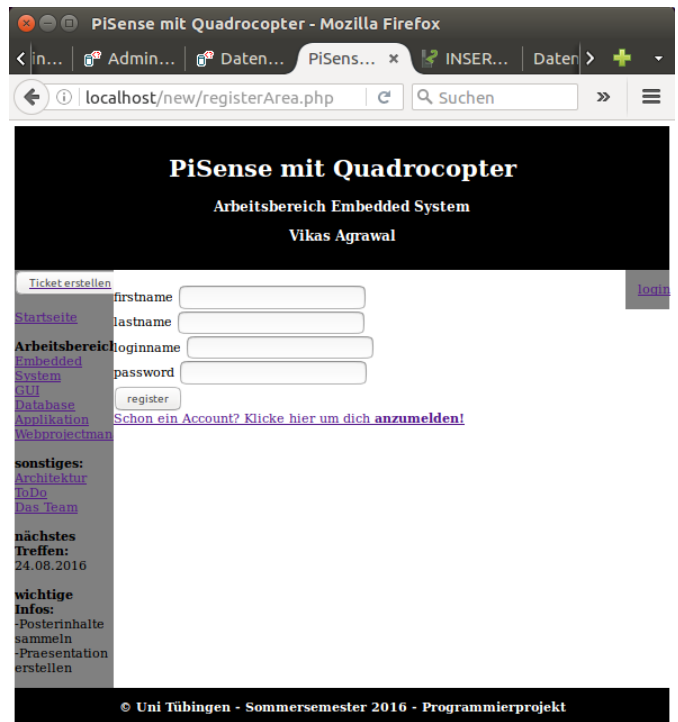


Bild: Loginbereich

Die Loginfunktion funktioniert wie folgt: Die Anmeldedaten werden über die GET methode in der Datei loginArea.php an die login.php geschickt und dort mit Daten aus der Datenbank verglichen. Wenn der loginname und das Passwort übereinstimmen, dann wird eine session gestartet und eine Sessionvariable gesetzt, welche speichert ob man angemeldet ist. Wenn alles erfolgreich war, dann wird man auf die index.php seite weitergeleitet via `header("Location: index.php")`. Dort sieht man nun auf der oberen rechten Seite statt dem login button einen logout button. Falls der login erfolglos war, wird man wieder zu loginArea.php Seite weitergeleitet und der login Button auf der oberen rechten Seite ist bei login.

```
loginArea.php (~\Schreibtisch) - gedit
Öffnen  Speichern
< solveTicket.php x  todo.php x  loginArea.php x  login.php x >
<?php
<doctype html>
<head>
    <header>
        <h1>PiSense mit Quadrocopter</h1>
        <h3>Architektur des Projekts</h3>
    </header>
    <?php include ("header.php"); ?>
</head>
<body>
    <content>
        <br>
        <form action="login.php" method="GET">
            <label for="loginname"> loginname</label>
            <input type="text" name="loginname">

            <label for="password"> password</label>
            <input type="password" name="password">

            <input type="submit" value="login">

        </form>

        <a href="registerArea.php" title="Noch kein Account? Hier klicken zum registrieren!"> Noch kein Account? Hier klicken zum registrieren! </a>

    </content>

    <?php include ("footer.php"); ?>
</body>

PHP  Tabulatorbreite: 8  Z. 1, Sp. 16  EINF
```

Bild: Quellcode loginArea.php

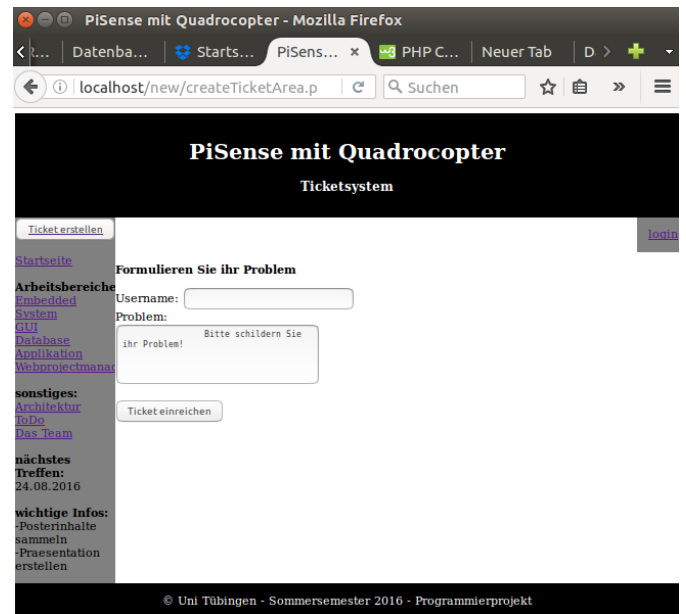


Bild: Quellcode createTicketArea.php

zukünftige Ziele:

Real Time Code Generation mit MatLab Simulation

```
*logout.php [Schreibgeschützt] (/var/www/html/new) - gedit
Öffnen  Speichern
< solveTicket.php x  todo.php x  loginArea.php x  login.php x  *logout.php x >
<?php
session_start();
$_SESSION['login'] = false;
session_destroy();
header("Location: index.php");
?>
```

Bild: Quellcode logout.php

Das Ticketsystem:

Das Ticketsystem wurde entwickelt damit Teammitglieder im Projekt ein Ticket erstellen können und so ihre Fragen und Probleme in der Gruppe stellen. Teammitglieder welche ein bisschen Zeit übrig haben oder sich Zeit schaffen, haben dann die Möglichkeit das Ticket zu bearbeiten.