**Projekt Drohne**

**Gruppenmitglieder**

Klaudia Paradowska

Ines Marquardt

Elmas Ahsen Ünal

Dilek Aydin

(Dr. rer. Klauth, Tim Schmitz)

**Projektbeschreibung**

Die Drohne soll aus der Luft Fotos Machen und diese an die Bodenstation schicken.

Während des Fluges soll die Kommunikation zwischen Bediener und Drohne möglich sein.

Anschließend soll es sicher auf Boden landen.

**Verwendete Bauteile**

Raspberry Pi 3

Pixy-Cam

Pi-Kamera

W-Lan-Stick

Leuchtbirne

Drohne mit Flight-Controller

**Inhaltsverzeichnis**

1. **W-Lan**
   1. W-LAN über supplicant
   2. W-LAN über einen Wi-Fi-Adapter
2. **Pi zu Putty**
   1. Verbindung zum Monitor
3. **Pixy-Cam – PixyMon**
   1. Pixy-Cam
   2. PixyMon 2.0.9
   3. Installieren und verbinden auf einem Windows-Rechner
   4. Installieren und verbinden auf dem Raspberry-Pi
   5. Fokus einstellen
   6. Farbsignatur anlernen
   7. Anlernen ohne PixyMon
4. **Pi-Kamera**
5. **Leuchtbirne**
6. **Skizze**
7. **Apache Server Installation**
8. **Webseite erstellen**
   1. Webseitengröße bestimmen
   2. Header
   3. Navigationszeile
   4. Fußzeile
9. **HTML-/PHP-Seite**
10. **Kommunikation zwischen Drohne und Webseite**
    1. Start-Button
    2. Stopp-Button
    3. Pixy-Cam erkennt ein Objekt
    4. Pixy-Cam erkennt kein Objekt
11. **Legende**
12. **Quellen**

**1. W-Lan**

**1.1. W-LAN über supplicant**

Für das W-LAN benötigen wir zuvor einige kleine Einstellungen. Wir ändern unter /etc/network/interfaces ein Paar Einstellungen. Hierzu geben wir ein:

sudo nano /etc/network/interfaces

Und verweisen auf /etc/wpa\_supplicant/wpa\_supplicant.conf, wo wir unsere Verbindungsdaten eintragen. Die Datei wpa\_supplicant.conf muss nun ebenfalls bearbeitet werden. Dort tragen wir den Netzwerknamen ein, in welches wir uns einwählen möchten und hinterlegen dort auch den W-LAN-Schlüssel. Nach einem Neustart verbindet sich der Raspberry automatisch in das konfigurierte Netzwerk.

**1.2. W-LAN über einen Wi-Fi-Adapter**

Um den Raspberry-Pi W-Lan fähig zu machen, müssen wir den Adapter EW-7833 UAC auf Raspberry-Pi konfigurieren. Dies geschieht folgendermaßen:

*Optionales Update: sudo apt update*

*sudo apt upgrade*

*sudo apt dist –upgrade*

*sudo reboot*

Treiberinstallation: *sudo apt install –build-essential git bc*

Header-Date Installation:

*git clone https://github.com/notro/rpi-source.git*

*cd rpi source*



*chmod +x rpi-source*

*sudo apt install libncurses5 –dev*

*sudo ./rpi-source*

*cd ..*

*git clone https://github.com/diederikdehaas/rtl8814AU.git*

*cd rtl8814AU/os\_dep/linux*

*sudo nano usb\_intf.c*

nach Zeile #212*: {USB\_DEVICE (ox7392. OxA833), driver\_info = RTL8814A}, /\*Edimax – Edimax\*/*

*speichern*

*cd rtl8814AU*

*sudo nano Makefile*

Zeile #921 *ARCH?=$/SUBARCH* ändern zu *ARCH=arm*

*speichern*

Kompilieren, Treiber installieren:

*sudo make*

*sudo make install*

System neu starten: *sudo reboot*

**2. Pi zu Putty**

Verbindung zum Monitor ist möglich über Putty. Zuvor muss das Programm Putty installiert sein. Einfacher und schneller geht die Verbindung per HDMI-Kabel zum Monitor. Unter Umständen ist die serielle Schnittstelle des Raspberry-Pi nicht aktiv. Hierzu muss man die folgenden Anweisungen befolgen, um die Schnittstelle zu aktivieren.

*raspi-config*

🡪 Advance Options -> Serial -> Ja --> „Enable“

Anschließend kann man via *startx* die grafische Benutzeroberfläche starten. Danach kann man das W-Lan manuell anschalten.

**3. Pixy-Cam - PixyMon 2.0.9**

**3.1. Pixy-Cam**

- 50 Bilder pro Sekunde

- Für alle Bibliotheken: Raspberry-Pi, Arduino etc.

- Programmiersprachen: Python, C / C ++

- Schnittstellen: Serial, SPI, USB, Digital/Analog, UART, I2C

- Blickwinkel: 75° horizontal, 47° vertikal

- Kosten: ca. 105 €

**3.2. PixyMon 2.0.9**

Software zur Verwendung der Pixy-Cam-Kamera auf dem Raspberry-Pi oder Windows-Rechner.

**3.3. Installieren und verbinden auf einem Windows-Rechner:**

Unter cmucam.org/projects/cmucam5/wiki/Latest\_release PixyMon Version 2.0.9 für Windows installieren. Im Explorer PixyMon/bin/PixyMon.exe öffnen. Pixy-Cam mit einem Mini-USB-Kabel und dem Rechner verbinden.

**3.4. Installieren und verbinden auf dem Raspberry-Pi:**

Verbindung:

****Pixy-Cam mit einem seriellen Mini-USB-Kabel an den Raspberry Pi anschließen.

Installation:

*sudo apt-get install git*

*sudo apt-get install libusb-1.0-0-dev*

*sudo apt-get install qt4-dev-tools*

*sudo apt-get install qt4 –qmake*

*sudo apt-get install qt4 –default*

*****sudo apt-get install gt4*

*git clone https://github.com/charmedlabs/pixy.git*

*cd pixy/scripts*

*./build\_pixymon\_src.sh*

*cd ../src/host/linux*

*sudo cp pixy.rules /etc/udev/rules.d/*

*cd ../../../build/pixymon/bin/*

*./PixyMon*

Optional: *sudo ./install\_libpixyusb.sh*

PixyMon öffnen: *cd pixy/build/pixymon/bin*

*./PixyMon*

**3.5. Fokus einstellen**

Um die Kamera etwas schärfer zu stellen dreht man die Linse so weit auf, bis das Programm ein schärferes Bild zeigt. Die Auflösung der Kamera ist ziemlich gering, weswegen das Bild etwas verpixelt ist.

**3.6. Farbsignatur anlernen**

Die Kamera Pixy-Cam kann sich bis zu 7 Farbsignaturen merken. Zuerst positioniert man das Objekt vor der Kamera. Dann klickt man in der Menüleiste auf Action->Set Signature 1: Das Bild wird eingefroren, sodass man ein Rechteck um die erkannte Farbfläche ziehen kann. Danach wird die Farbe erkannt und unter s=1 gespeichert.

**3.7. Anlernen ohne PixyMon**

Es gibt die Möglichkeit Objekte ohne PixyMon anzulernen. Dazu drückt man den weißen Taster auf der Kamera solange, bis die LED rot leuchtet. Dann hält man das Objekt vor die Kamera und kann anhand der LED Farbe, die erkannte Farbe sehen. Anschließend betätigt man nochmal den Taster und die Signatur wird gespeichert. Leider, funktioniert diese Methode nur beschränkt, weshalb man die Signaturen lieber über PixyMon anlernen sollte.

**4. Pi-Kamera**

Verbinden der Pi-Kamera mit dem Raspberry-Pi.

Über Windows: 🡪 Menü -> References -> Raspberry Pi Configuration ->

Interfaces -> Camera--> „Enable“

Über Rasperry-Pi: sudo *raspi-config*

6 Enable Camera -> Enable – Yes

Foto machen: *raspistill -o image.jpg*

Das gemachte Foto kann man über den Explorer auf der Bedienoberfläche ansehen.

Video machen: *raspivid -o video.h264 -t 10000*

Die 10000 am Ende geben die Dauer des Videos an. Hierbei ist die Angabe in Millisekunden zu machen.

Video öffnen: *omxplayer video.h264*

**5. Leuchtbirne**

Die Leuchtbirne wird mit dem Raspberry-Pi verbunden. Hierzu benötigen wir ein Akku. Mittels eines Python-Programms sorgen wir dafür, dass die Birne leuchtet, sobald die Pixy-Cam ein Objekt erkennt.  
Dafür benötigen wir im Programm-Code die Implementierungen RPi.GPIO und Time, die wir als Pakete importieren.

*while 1:*

*count = pixy\_get\_blocks(100, blocks)*

*if count > 0:*

*print ("frame %3d:' % (frame)")*

*frame = frame + 1*

*for index in range (0, count):*

*print ("[BLOCK\_TYPE=%d SIG=%d X=%3d Y=%3d WIDTH=%3d HEIGHT=%3d]' % (blocks[index].type, blocks[index].signature, blocks[index].x, blocks[index].y, blocks[index].width, blocks[index].height)")*

*GPIO.output(40, GPIO.HIGH)*

*time.sleep (1)*

*GPIO.output(40, GPIO.LOW)*

**6. Skizze**

****

Flightcontroller

Drohne

Pixy-Cam

Raspberry-Pi

Pi-Kamera

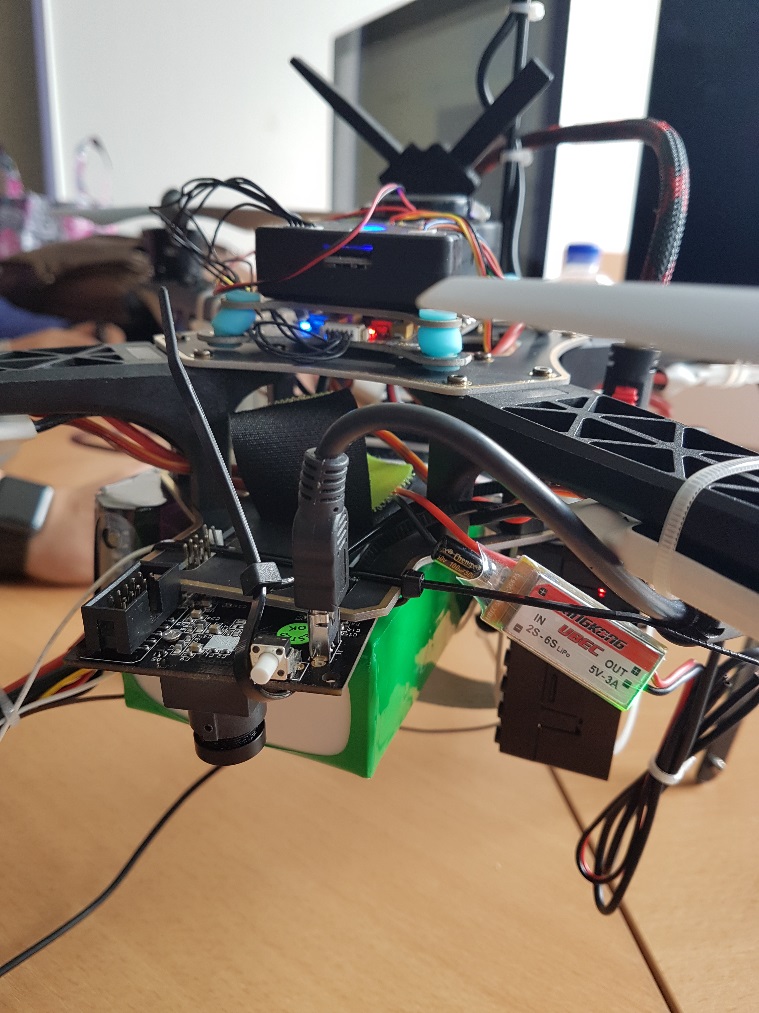
GPS

Stromversorgung

Wi-Fi-Adapter

Leuchtbirne

Akku für die Lampe

****

Flightcontroller

Akku für die Lampe

Raspberry-Pi

Pixy-Cam

Pi-Kamera

Stromversorgung

**6. Apache Server Installation**

Der Apache Server wird für die Anwendung eines Webservers benötigt. Mit den folgenden Codes laden wir den beliebtesten Webserver und die Serversprache PHP auf unserem Raspberry-Pi:

*sudo apt-get install apache 2 php5*

*sudo /etc/init.d/apache 2 start oder stop*

*/var/www.index.html*

*cd /var/www*

*sudo nano index.html*

Anschließend meldet man sich mit einer ssh-Verbindung und der zuvor ermittelten IP-Adresse auf dem Raspberry-Pi an.

**7. Webseite erstellen**

Zur Gestaltung der Website wird eine CSS-Datei angelegt.

In der CSS-Datei legen wir zunächst die Hintergrundfarbe auf grau fest und setzen den Außenrandabstand auf 0. Durch den Befehl Front Family wird die Schriftart der Webseite festgelegt. Wir haben uns hier für Georgia entschieden. Da es sich aber um keine Standardschriftart handelt, haben wir Arial als zweite Schriftart gewählt. Diese wird erst benutzt, wenn die ersten beim Nutzer der Webseite nicht verfügbar ist.

*body*

*{*

*background-color: gray;*

*margin: 0;*

*front-family: Georgia, Arial;*

*}*

**7.1. Webseitengröße bestimmen**

Nun bestimmten wir den äußeren Abstand und die Breite der Webseite. Der Container ist auf eine Breite mit 900 Pixel eingestellt. Durch margin-left und margin-right wird der automatische Abstand nach links und rechts festgelegt. Dadurch kann die Webseite sich optimal auf die verschiedenen Endgeräte anpassen. Padding ist für einen erzwungenen Leerraum zwischen dem Inhalt eines Elements und seinem eigenen Elementrand. Hier setzen wir ihn auf 0. Nach oben ist ein Abstand von 15 Pixel festgelegt. Den Container haben wir mit deinen Rahmen versehen (border-radius).

Overflow:hidden legt fest, dass ein übergroßer Inhalt eines Elements beschnitten wird.

#wrapper

*{*

*width: 900px;*

*margin-left: auto;*

*margin-right: auto;*

*padding: 0;*

*margin-top: 15px;*

*border-radius: 15px;*

*overflow: hidden;*

*}*

**7.2. Header**

Im Bereich Header kümmern wir uns um die Gestaltung des Kopfes. Als erstes legen wir die Höhe des Kopfes mit 150 Pixel fest. Am Rand fügen wir danach das Logo der Hochschule Niederrhein ein.

*{*

*height: 150px;*

*background-image: url('hsnrlogo.jpg');*

*}*

**7.3. Navigationszeile**

Hier wird die Höhe der Navigationszeile festgesetzt mit 30 Pixel.

*nav*

*{*

*height: 30px;*

*}*

Als nächsten legen wir die Farbe der Schrift in der Navigationszeile fest (*color:white*). Durch den Befehl *text-decoration: none* können wir bestimmen, ob der Text unterstrichen oder fett gedruckt werden soll. In diesen Fall verhindern wir dadurch, dass die hinterlegten Links unterstrichen angezeigt werden.

In der Navigationszeile sollen die Elemente hintereinanderstehen. Dies lässt sich am einfachsten durch eine Liste gestalten. Allerdings sollen keine Aufzählungszeichen oder Nummerierungen an den Elementen vorhanden sein. Dies erreichen wir durch *list-style-type* auf *None* zu setzen.

Die Elemente sollen an den Links am Rand der Navigationszeile starten, was wir durch den Befehl *float: left* erreichen. Mit *margin* legen wir erneute die Außenabstände in der folgenden Reihenfolge fest: oben, rechts, unten, links.

In diesen Abschnitt legen wir fest, dass in der Navigationszeile die Links unterstrichen werden, sobald wir den Mauszeiger auf die dort hinterlegten Links bewegen. Dadurch wird für den Nutzer besser ersichtlich, wo er sich mit dem Mauszeiger befindet.

*nav a: hover*

*{*

*text-decoration: underline;*

*}*

**7.4. Fußzeile**

In der Fußzeile möchten wir, dass der Text mittig zu lesen ist (*text-align: center*).

Außerdem soll ein Strich die Fußzeile von dem Rest der Seite abtrennen (*border-top: 1px solid black*).

Den Leerraum legen wir mit 2 Pixeln fest.

**8. HTML-/PHP-Seite**

Zunächst müssen wir den Kopf unserer Webseite gestalten. Dafür legen wir als erstes den Titel mit Drohne Livestream fest.

Durch den ersten Meta Befehl erreichen wir, dass sich die Seite einmal pro Sekunde aktualisiert.

Unter Google kann mein unsere Webseite durch den Suchbegriff „Drohne eHealth“ finden. Dies haben wir durch *description content* bestimmt.

Zum Schluss verlinken wir unsere Webseite mit den nötigen *Stylesheet* und fügen *javascript* als Sprache ein.

*<head>*

*<title>Drohne Livestream</title>*

*<meta http-equiv="refresh" content="1">*

*<meta name="description" content="Drohne eHealth" />*

*<meta charset="UTF-8" />*

*<link rel="stylesheet" type="text/css" href="style.css" />*

*<script language="javascript" type="text/javascript" src="script.js"></script>*

*</head>*

Hier wird die Navigationszeile unserer Seite gestaltet. Wir verlinken unsere Startseite und unseren Studiengang.

<*nav>*

*<ul>*

*<li><a href="index.html">Startseite</a></li>*

*<li><a href="https://www.hs-niederrhein.de/services/studieninteressierte/studienangebot/studiengang/b-sc-ehealth-it-im-gesundheitswesen">eHealth Studium</a></li>*

*</ul>*

*</nav>*

Nun fügen wir die ersten Texte auf der Webseite ein.

*<div id = "content">*

*<h1> Willkommen </h1>*

*<p> auf unserer Seite. </p>*

*<p> Hier stellen wir euch unsere Projektarbeit vor. </p>*

*</div>*

Nun geht es weiter mit dem PHP Teil. Es wird mit der Schleife geprüft, ob eine Datei im Ordner Home/pi/drohne abgelegt wurde. Wenn eine Datei gefunden wurde, wird auf der Webseite „Objekt gefunden“ angezeigt:

*<?php*

*if(file\_exists("../home/pi/Drohne/Huhu"))*

*echo "Objekt gefunden";*

Falls kein Objekt gefunden, wurde tritt else ein.

else

*echo "Objekt nicht gefunden";*

Es wird dann „Objekt nicht gefunden“ ausgegeben.

Nun fügen wir zwei Buttons mit der Aufschrift Start und Stopp ein. Die Datei index.php reagiert auf die Erkennung des Pixy-Cams. Wenn die Pixy-Cam ein Objekt erkennt so leuchtet die Leuchtbirne und man kann aus dem Browser durch bestätigen des Stopp-Buttons die Drohne in den „Hold“-Zustand versetzen. Über den Start-Button fliegt die Drohne weiter.

*<form method="get" action="index.php">*

*<input type="submit" value="Start" name="Start">*

*<input type="submit" value="Stop" name="Stop">*

*</form>*

Zum Schluss fügen wir eine Fußzeile ein. Hier kann später an der Stelle von # der Link für ein Impressum hinterlegt werden.

*<footer>*

*<p> <a href = "#"> Impressum </a> </p>*

*</footer>*

**9. Kommunikation zwischen Drohne und Webseite**

Die Kommunikation zwischen der Drohne und der Webseite wird über ein Python-Programm geregelt. Bei der Kommunikation geht es darum, ob die Pixy-Cam aus der Luft ein Objekt (Farbe) erkennt. Wenn die Pixy-Cam ein Objekt erfasst hat, wird die Pi-Kamera ein Foto machen, und dieses auf der Webseite darstellen. Solange soll die Drohne im „Hold-Modus“ in der Luft stehen. Ist auf dem Foto ein Objekt dargestellt, wonach gesucht wird, gibt es zwei Möglichkeiten. Erstens können wir der Drohne durch Anklicken des Start-Buttons mitteilen, dass sie weiterfliegen soll. Zweitens kann man durch Anklicken des Stopp-Buttons mitteilen, dass sie zurückfliegen soll.

Damit dies gelingt, wird ein Ordner auf dem Raspberry-Pi erstellt, worauf die Webseite zugreift und der Raspberry-Pi durch ein Python-Programm ebenfalls ständig abfragen in diesem Ordner durchführt.

**9.1. Start-Button**

In diesem Ordner schreibt die Webseite über das PHP-Skript eine Datei mit dem Inhalt 1, wenn auf Start geklickt wird. Das Python-Programm schaut in diese Datei rein und erkennt die 1. Die 1 stellt eine Anweisung für den Flightcontroller dar, die Suche zu starten. Der Flightcontroller geht über in den Modus „Auto“. In diesem Modus fliegt die Drohne und scannt die Gegend über die Pixy-Cam ab.

**9.2. Stopp-Button**

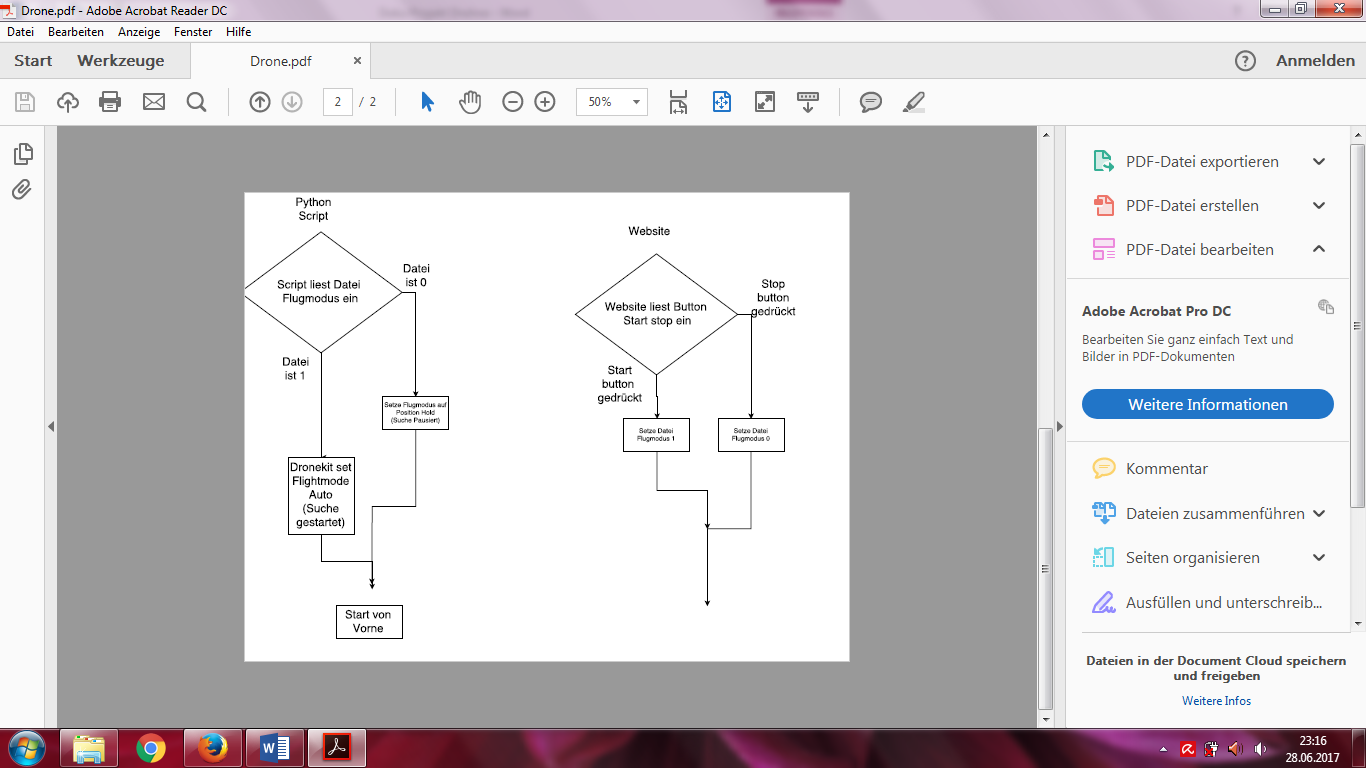
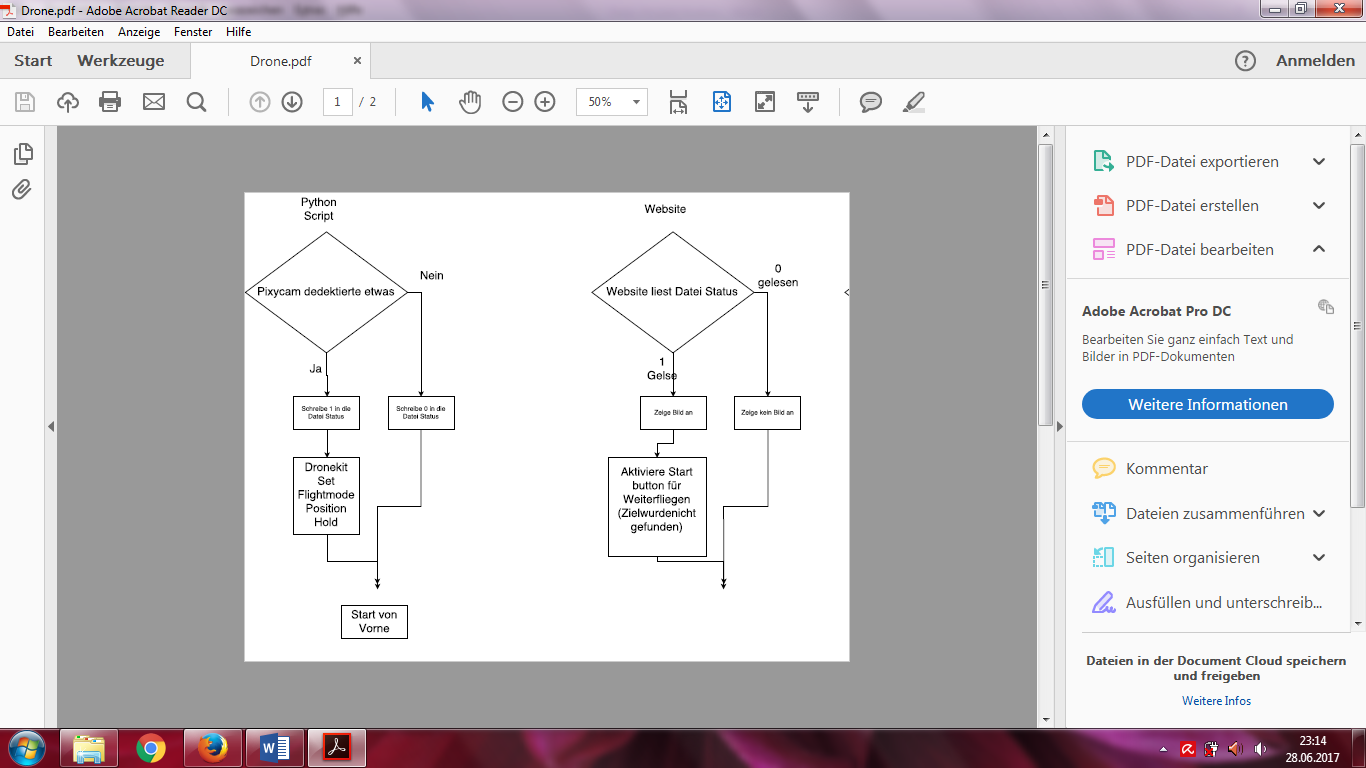
Wenn auf der Webseite der Stopp-Button geklickt wird, schreibt das PHP-Skript in die Datei eine 0. Sobald der Raspberry-Pi durch die Abfrage des Python-Programms entdeckt, gibt er die Anweisung an den Flightcontroller, dass er die Suche unterbricht und auf den Flugmodus „Attitude Hold“ wechselt. In diesem Modus bleibt die Drohne an Ort und Stelle.

**9.3. Pixy-Cam erkennt ein Objekt**

Sobald die Drohne ein Objekt erkennt, schreibt das Python-Programm in die Datei eine 1 und ändert den Flugmodus auf „Attitude Hold“. Wir sehen das Bild auf der Webseite und können über Start und Stopp die Drohne weiterfliegen lassen oder zurückkommen lassen.

**9.4. Pixy-Cam erkennt kein Objekt**

Findet die Drohne kein Objekt, schreibt das Python-Programm in die Datei eine 0. Solange in der Datei eine 0 steht, fliegt die Drohne weiter und wir können die Suche über Stopp anhalten.



**10. Legende:**

*Kursiv* = Befehlseingaben im Editor

**11. Quellen:**

<https://edimax.freshdesk.com/support/solutions/articles/14000057252-how-to-install-ew-7833uac-adapter-in>-linux-running-kernel-higher-than-v4-4

<https://elv.de/pixy-cam-fuer-minicomputer-wie-raspberry-pi-arduino-und-co.html>

[https://www.conrad.at/de/kameramodul-pixy-cam-passend -fuer-arduino-boards-raspberry-pi-banana-pi-pcduino-1364834.html](https://www.conrad.at/de/kameramodul-pixy-cam-passend%20-fuer-arduino-boards-raspberry-pi-banana-pi-pcduino-1364834.html)

**Quellen für die HTML und CSS Seite**

<https://www.youtube.com/watch?v=j-6eOHxtd1M>

<https://www.youtube.com/watch?v=fItb2xzsZrI&t=15s>

<https://www.youtube.com/watch?v=65cI1SN0isg>

**Quellen für PHP**

Unterlagen aus dem ersten Semester: Skript zum Raspberry-Pi Teil 2

<https://www.php-einfach.de/php-tutorial/if-anweisungen/>

<http://www.phpbox.de/php_befehle/file_exists.php>

<http://php.net/manual/de/control-structures.elseif.php>

<https://www.php-kurs.com/variablen-in-php-nutzen.htm>

<https://www.w3schools.com/php/php_syntax.asp>