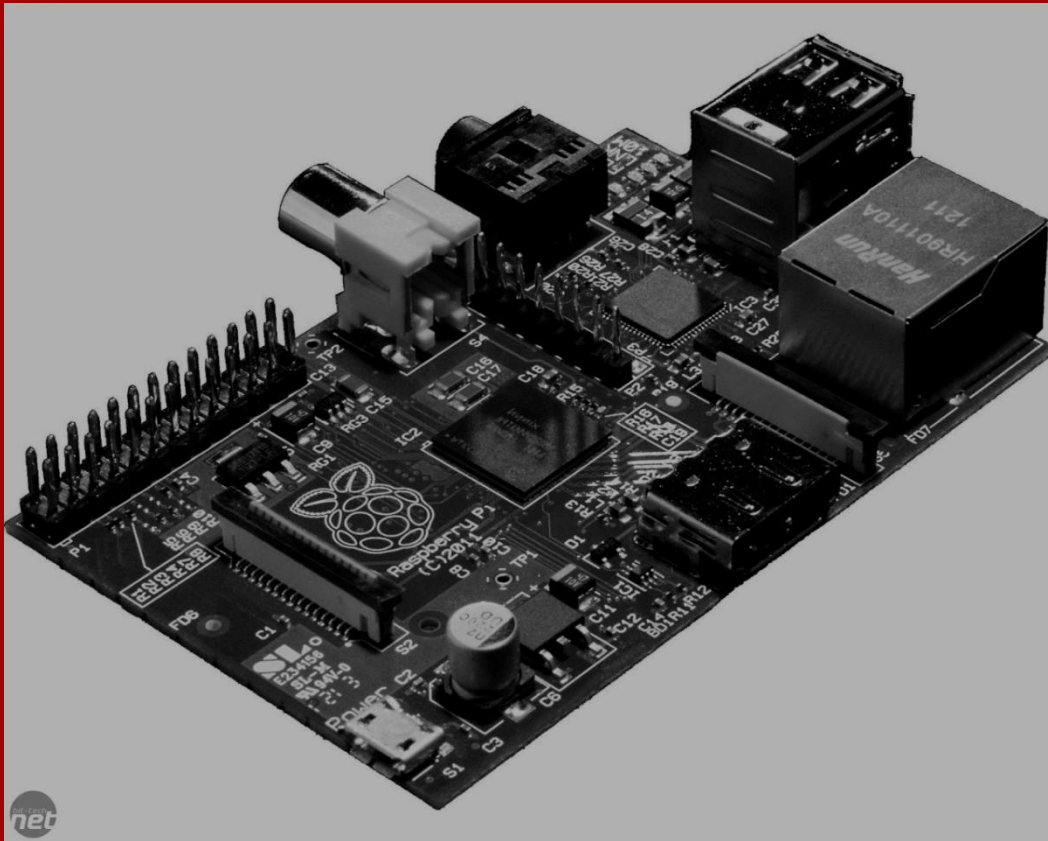


18.09.2014
5AHWI



PFLICHTENHEFT EINLESEGERÄT

Betreuungslehrer: PROF. KÖLLÖ SZABOLCS,

CARPENTARI LUKAS
FODOR JULIAN

INHALTSVERZEICHNIS

| | |
|--|---|
| 1. Zielbestimmungen..... | 1 |
| 1.1 Muss-Kriterien..... | 1 |
| 1.2 Kann-Kriterien | 1 |
| 1.3 Abgrenzungskriterien | 1 |
| 2. Produkteinsatz | 1 |
| 2.1 Anwendungsbereich..... | 1 |
| 2.2 Zielgruppe..... | 1 |
| 2.3 Betriebsbedingungen | 2 |
| 3. Produktumgebung..... | 2 |
| 3.1 Software | 2 |
| 3.2 Hardware | 2 |
| 4. Produktfunktionen | 2 |
| 5. Produktdaten | 2 |
| 5.1 Gespeicherte Zählerstände | 3 |
| 6. Qualitätszielbestimmungen | 3 |
| 7. Globale Testszenarien und Testfälle | 3 |
| 7.1 Schritte der Testreihe:..... | 3 |
| Schritt1: | 3 |
| Schritt2: | 3 |
| Schritt3: | 3 |
| 8. Entwicklungsumgebung | 3 |
| 8.1 Software | 3 |
| 8.2 Hardware | 4 |
| 8.3 Orgware..... | 4 |
| 9. Kostenstellen..... | 4 |
| 10.GANTT-PLAN..... | 5 |

EINLESEGERÄT

1. ZIELBESTIMMUNGEN

Es soll ein Gerät entwickelt werden, welches den Zählerstand eines Stromzählers, Wasserzählers oder eines Wärmezählers einlesen kann. Des Weiteren soll dieses Gerät die eingelesenen Daten direkt auf einem USB-Stick speichern.

1.1 MUSS-KRITERIEN

- Es sollen Zahlen mit einem Bilderkennungs-Programm erkannt und unterschieden werden.
- Das Einlesegerät soll für gängige Zählertypen funktionieren.
- Jede Minute soll der Zählerstand ausgelesen werden.
- Die Zählerstände müssen auf einen USB-Stick in ein Textdokument gespeichert werden.

1.2 KANN-KRITERIEN

- Die Daten sollen direkt in einem Diagramm ausgewertet werden.
- Die Daten werden auf eine Website geladen und sind dort abrufbereit.
- Es soll ein „Dropbox-System“ zwischen den verschiedenen Raspberry Pi entwickelt werden.
- Es soll möglich sein die Einlese Prozesse zu pausieren.
- Die Daten sollen in eine SQLite Datenbank gespeichert werden.

1.3 ABGRENZUNGSKRITERIEN

- Das System muss nur für handelsübliche Zähler funktionieren.
- Das System soll nur Zahlen erkennen.

2. PRODUKTEINSATZ

2.1 ANWENDUNGSBEREICH

Personen verwenden das Produkt um bequemer Zählerstände ablesen und auswerten zu können. Des Weiteren ist es möglich anhand der aufgenommenen Daten Statistiken zu entwerfen, da jede Minute der Zählerstand gespeichert wird.

2.2 ZIELGRUPPE

Zur Zielgruppe zählen Menschen die für die Instandhaltung eines Unternehmens zuständig sind. (z.B Hausmeister). Das Produkt kann aber auch für den Privathaushalt verwendet werden.

Es werden Basiskenntnisse in der Verwendung mit USB-Sticks vorausgesetzt.

Des Weiteren sollten zumindest geringe technische Kenntnisse vorhanden sein im Bereich des Energiehaushaltes von Gebäuden.

2.3 BETRIEBSBEDINGUNGEN

- Betriebsdauer: täglich 24 Stunden
- Leicht zu Warten und nicht Fehleranfällig
- Die Entleerung des USB-Sticks muss manuell vorgenommen werden.

3. PRODUKTUMGEBUNG

Das System arbeitet größtenteils unabhängig. Lediglich zum Übertragen der Daten vom USB-Stick auf ein externes Gerät wird ein weiteres Speichermedium benötigt.

3.1 SOFTWARE

- Programm zum Auswerten der Bilder
- Programm welches die Daten in eine Textdatei speichert

3.2 HARDWARE

- Medium welches Bilder schießt
- Speichermedium
- Mini-PC
- Beleuchtung

4. PRODUKTFUNKTIONEN

Das Produkt nimmt dem Benutzer die Arbeit ab, sich die Zählerstände immer wieder notieren zu müssen und versorgt diesen mit zusätzlichen Informationen zu dem Verlauf des Ressourcenaufwandes, da jede Minute ein Bild (Zählerstand) abgespeichert wird.

Wir versuchen die Lösung auf zwei verschiedenen Wegen zu erreichen, indem wir zwei verschiedene Open source Produkte verwenden. Das erste Produkt ist die OpenCV library und das zweite Produkt ist die Google tesseract Library.

5. PRODUKTDATEN

Gespeichert werden alle Bilder die aufgenommen wurden und die dadurch generierten Zahlen.

5.1 GESPEICHERTE ZÄHLERSTÄNDE

Jedes geschossene Bild soll mit einem Programm so bearbeitet werden, dass man die Zählerstände in eine Textdatei speichern kann und diese dann mit einem USB Stick abholen kann.

6. QUALITÄTSZIELBESTIMMUNGEN

| | Sehr wichtig | wichtig | unwichtig |
|------------------------------|--------------|---------|-----------|
| Zuverlässigkeit | x | | |
| Leicht zu Bedienen | | | x |
| Effizient | | x | |
| Korrektheit der Daten | x | | |
| Wartungsarm | | | x |

7. GLOBALE TESTSZENARIEN UND TESTFÄLLE

Das Produkt wird direkt in der HTL an Stromzähler, Wärmezähler und Wasserzählern getestet.

7.1 SCHRITTE DER TESTREIHE:

SCHRITT1:

Es sollen Bilder von Zahlen gemacht werden und anschließend soll überprüft werden ob die Zahlen richtig eingelesen wurden.

SCHRITT2:

Das Produkt wird eine Woche am Zähler montiert.

SCHRITT3:

Der USB-Stick wird dem Ablesegerät entnommen und es wird geprüft ob die Daten abgespeichert wurden.

8. ENTWICKLUNGSUMGEBUNG

8.1 SOFTWARE

- C++
- OpenCV library
- tesseract
- Linux

8.2 HARDWARE

- Raspberry Pi
- USB-Stick
- Kamera für Raspberry Pi
- LEDs
- Hardware-Uhr

8.3 ORGWARE

- Zähler

9. KOSTENSTELLEN

| Utensilien | Kosten |
|---------------------------|--------------------------------|
| Raspberry pi | Gestiftet von Prof. Frischmann |
| LEDs (30 Leuchtdioden) | 3€ |
| Kamera | Gestiftet von Prof. Frischmann |
| USB-Stick (16 GB) | 8€ |
| MicroSD – Card (16GB) | 10€ |
| Ethernet-Kabel(Crossover) | 6 € |
| Hardware-Uhr | 2€ |
| Summe | 29€ |

10.GANTT-PLAN

| Vorgang | Anfang | Ende |
|--|----------|----------|
| • Einführung | 18.09.14 | 07.10.14 |
| ♀ • Recherche/Prototyp | 08.10.14 | 15.12.14 |
| • Recherche über Linux/Prototyp | 08.10.14 | 27.10.14 |
| • Recherche über Raspberry PI/Prototyp | 08.10.14 | 05.11.14 |
| • Recherche C++/Prototyp | 06.11.14 | 15.12.14 |
| • Recherche Bilderkennung/Prototyp | 06.11.14 | 04.12.14 |
| ♀ • Programmierung | 16.12.14 | 03.03.15 |
| • Einarbeiten in C++ | 16.12.14 | 02.01.15 |
| • Tests mit C++ Libraries Tesseract | 05.01.15 | 02.02.15 |
| • Bilderkennung Programmieren | 05.01.15 | 11.02.15 |
| • Ausgabe auf USB Stick Programmieren | 05.01.15 | 02.02.15 |
| • Programmevaluierung | 12.02.15 | 03.03.15 |
| • Testen | 04.03.15 | 23.03.15 |
| ♀ • Evaluierung | 24.03.15 | 22.04.15 |
| • Fehler ausbessern | 24.03.15 | 10.04.15 |
| • Programm endgültig abschließen | 13.04.15 | 21.04.15 |
| • Projekt Abschließen | 22.04.15 | 22.04.15 |

