Pflichtenheft

Visuelle Programmiersprache für den Physikunterricht zur Datenerfassung auf einem Raspberry Pi

Version 0.2.2

David Gawron Stefan Geretschläger Leon Huck
Jan Küblbeck Linus Ruhnke

26. Mai 2019

Inhaltsverzeichnis

1 Produktübersicht

Die Anwendung soll es Lehrern ermöglichen Schülern die Grundlagen von physikalischen Messtechniken zu vermitteln (siehe Kapitel 2). Demzufolge kommt die Anwendung in Schulen zum Einsatz. Die Zielgruppe sind Schüler in der 7. Klasse aufwärts. Die Anwendung läuft auf einem Computer und erhält Messwerte von Sensoren. Die Daten werden durch PhyPiDAQ verfügbar gemacht. Detaillierte funktionale und nicht funktionale Anforderungen sind in Kapitel 5 bzw. 7 aufgelistet.

2 Zielbestimmung

Die Anwendung soll es Lehrern ermöglichen, Schülern ab der siebten Klasse Grundkenntnisse der digitalen Messwerterfassung in einer für Schüler interessante und motivierenden Weise näher zu bringen. Dabei werden aus didaktischer Sicht überflüssige technische Details wie z. B. die zum Auslesen der Sensoren notwendigen Protokolle vor dem Schüler verborgen. Der Schüler wird so nicht überfordert, sondern soll ermutigt werden, selbstständig mit der Software umzugehen. Dabei kann er im spielerischen Umgang mit Versuchsaufbauten Prinzipien der digitalen Messtechnik wie z. B. Kaskadierung und natürlich auch das Grundprinzip von Ursache und Wirkung erfahren.

Es wird eine graphische Oberfläche angeboten, die es dem Schüler ermöglicht, allein per Drag and Drop eine Messanordnung zu erstellen. Überflüssige Details, die dahinter stecken, bleiben vor dem Schüler verborgen. Die Anwendung motiviert den Schüler dazu, mit Sensoren, Transformationen und Darstellungen zu spielen und Dinge auszuprobieren. Dabei wird ihm durch eine intuitive Status- und Fehleranzeige gezeigt, ob seine Konstruktion funktioniert. Wenn nicht, dann zeigt sie ihm an, wo das Problem liegt und warum es nicht funktioniert. Eine lästige Fehlersuche soll dem Schüler weitestgehend erspart bleiben.

Außerdem liefert die Anwendung dem Schüler zu den vorhandenen Bausteinen und zu der Anwendung allgemein die nötigen Informationen, die er für die Nutzung braucht. Dabei wird auf ein ausführliches Tutorial am Anfang verzichtet. Die Anwendung liefert die Informationen häppchenweise durch Informationsanzeigen an den jeweilig relevanten Stellen. Damit findet der Schüler die Hilfe, die er sucht, an der Stelle, an der er sie braucht.

Die Anwendung ermöglicht es dem Lehrer vor dem Unterricht, eine Reihe von Messversuchen teilweise oder ganz zu konfigurieren und zu speichern. Dabei kann er genau bestimmen, was er den Schülern zeigen will. Diese Konfigurationen kann er dann schnell und einfach im Unterricht zum Einsatz bringen.

Weiter ermöglicht es die Anwendung, dass Schüler aus höheren Klassen mit mehr Details der digitalen Messwerterfassung zusammen gebracht werden. Diese nutzen keine oder nur eine abstrakt bzw. lückenhaft vorgefertigte Konfiguration. Sie können dann auch selbst einen Baustein modifizieren oder erstellen. Trotz des höheren Detailgrads bleibt die Anwendung übersichtlich und strukturiert. Damit bleiben auch komplexere Messversuche für den Schüler motivierend.

Die Anwendung ist als Bindeglied zwischen PhyPiDAQ und dem Nutzer zu verstehen. Sie ermöglicht dem Nutzer, über eine übersichtliche, strukturierte und intuitive GUI sowie über eine einfache Bedienung die Nutzung eines Raspberry Pi mit PhyPiDAQ. Dadurch soll dem Schüler digitale Messwerterfassung, Physik und auch Informatik in einer motivierenden Weise näher gebracht werden. Womöglich kann die Anwendung den Schüler auch für diese Themen begeistern.

Im Folgenden werden Musskriterien, Sollkriterien und Wunschkriterien des projektierten Softwareprodukts definiert:

- Musskriterien haben höchste Priorität. Dass ein Musskriterium in den nachfolgenden Projektphasen nicht umgesetzt wird, ist nur dann zulässig, falls unerwartet unausweichliche Probleme bei der Umsetzung auftreten. In diesem Fall ist es erforderlich, dass diese Probleme sehr genau dokumentiert werden.

- Sollkriterien haben eine mittlere Priorität. Falls ein Sollkriterium umgesetzt werden kann, dann muss es auch realisiert werden. Falls ein Sollkriterium in den nachfolgenden Projektphasen nicht umgesetzt werden kann, so muss dies dokumentiert und begründet werden.

- Wunschkriterien haben eine niedrige Priorität. Je nach Resourcenlage können sie nach Bearbeitung aller Muss- und Kannkriterien umgesetzt werden. Falls ein Wunschkriterium nicht umgesetzt werden kann, so muss dies nicht begründet werden.

2.1 Musskriterien

- Datenhandhabung
  - Der Benutzer kann Ergebnisse aus einer Messung speichern und laden.
  - Der Benutzer kann gleichzeitig Daten aus einer Messung darstellen und diese Daten auch weiter verwenden.
  - Der Benutzer kann eine Messkonfiguration speichern und laden.

  - Der Benutzer kann die Anwendung nutzen, ohne persönliche Daten preiszugeben.

- Benutzbarkeit der GUI

  - Die Anwendung reagiert auf Veränderungen so schnell, dass das Prinzip von Ursache und Wirkung intuitiv erfahren werden kann.

  - Der Benutzer kann Komponenten einer Messkonfiguration per Drag and Drop hinzufügen.

  - Der Benutzer kann Komponenten intuitiv mit einander verbinden.

  - Die Anwendung lässt sich mit dem Wissen eines Schülers aus der siebten Klasse bedienen.

  - Die Anwendung lässt sich auch mit einer rot-grün-Schwäche komplett nutzen.

- Funktionen

  - Die Anwendung erhält Daten durch den Raspberry Pi mit PhyPiDAQ oder aus einer Datei und kann diese dann transformieren und darstellen.

  - Die Anwendung enthält vorgefertigte Standardelemente, wie z.B. to do

  - Die Anwendung enthält vorgefertigte Standardkonfigurationen.

- Information und Rückmeldung

  - Ist ein verwendeter Sensor nicht angeschlossen oder fehlerhaft, meldet die Anwendung dies über eine Pop-Up-Nachricht.

  - Die Anwendung enthält Erklärungen und Informationen zu den einzelnen Komponenten, d.h. zu den Sensoren, Transformationen und Darstellungen.

  - Die Anwendung enthält Erklärungen und Informationen zu den GUI-Bereiche, d.h. zu dem Auswahlbereich der Komponenten, zu dem Messversuchsbereich und zu dem Anzeigebereich.

  - Falls nicht alle Kanäle verbunden sind, meldet die Anwendung dies dem Benutzer mit einer Fehlermeldung, falls dieser versucht die Messung zu starten.

- Sonstiges

- Die Anwendung kommuniziert mit den Nutzer auf deutsch.
- Die Anwendung soll auf die Unterstützung neuer Sensoren erweiterbar sein.
- Die Anwendung hält die Datenschutzrichtlinie „to do“ ein.
- Die Anwendung hält die Schulrichtlinie „to do“ ein.

2.2 Sollkriterien

- Die Anwendung soll auf die Erstellung eigener Transformationen erweiterbar sein.
- Die Anwendung soll auf andere Sprachen erweiterbar sein.
- Die Anwendung soll allein auf dem Raspberry Pi laufen können.
- Farbkodierung der GUI-Elemente
- Einfache Erweiterbarkeit

  Einheitliches Interface für Sensoren

2.3 Wunschkriterien

- Übertragung der Messdaten über Netzwerkschnittstelle
- Spiele, „nachmachen“ von vorgegebenen Messergebnissen
- Ausführliche Beschreibung der physikalische Hintergründe

2.4 Abgrenzungskriterien

- Unterschiedliche Benutzerkonten sind nicht zu implementieren
- Personenbezogene Daten werden nicht gespeichert

3 Produkteinsatz

3.1 Anwendungsbereiche

Die Anwendung ist für die Verwendung in Schulklassen ab der 7. bis zu 10. Klasse für Schüler und Lehrer im Physikunterricht konzipiert. Ebenfalls soll die Anwendung in Physikprojekten, sowie Science Labs verwendet werden können. Als Open Source Projekt Im Rahmen des OSL 2 steht die Anwendung jedoch jedem Interessierten zur Verwendung und Weiterentwicklung zur Verfügung.

3.2 Zielgruppe

Die Zielgruppen sind hauptsächlich Schülerinnen und Schüler im Physikunterricht, welche die Anwendung verwenden, um erste eigene Messungen durchzuführen. Das Ziel ist es den Schülern erste Einblicke in Messtechniken und Zusammenhänge zwischen Ursache und Wirkung zu zeigen. Die Anwendung soll ebenfalls nicht nur von physikbegeisterten Schülern verwendet werden können, sondern auch für Schüler ohne ein großes Vorwissen in der Physik und Messtechniken eine interessante Plattform zu bieten, welche einfach zu verwenden ist.

3.3 Betriebsbedingungen

Die Anwendung läuft auf gewöhnlichen Schulcomputern, welche über eine USB-Schnittstelle mit einem RasPi verbunden sind, auf welchem das Programm PhyPiDAQ läuft. Über das PhyPiDAQ werden die an das RasPi angeschlossenen Messsensoren adressiert, welche die Messdaten wiederum über das RasPi an die am Computer gelaufende Anwendung schicken.

4 Produktumgebung

Die Anwendung läuft auf einem Computer, die Messdaten werden über Messsensoren an einem Raspberry Pi erfasst.

4.1 Software

Die Anwendung läuft auf Computern mit den Betriebssystemen Linux ab Kernel 7 und Microsoft Windows ab Windows 7 oder neuer. Die Anwendung muss auf dem Computer vollständig installiert sein.

4.2 Hardware

Die Anwendung läuft auf gewöhnlichen Computern. Auf dem per USB-Schnittstelle verbundenen Raspberry Pi muss PhyPiDAQ installiert und verwendungsfähig sein. Die an den Raspberry Pi angeschlossenen Messsensoren müssen richtig und sinnvoll angeschlossen sein.

4.3 PhyPiDAQ

Bei PhyPiDAQ[1] handelt es sich um eine Anwendung zur Datenerfassung und Analyse mit einem Raspberry Pi. Diese ist nicht Bestandteil des Produktes, wird jedoch zur Datenerfassung und Datenverarbeitung und somit zur Funktionalität der Anwendung benötigt. Das Programm, welches in der Programmiersprache python3 geschrieben ist bietet einfache und einheitliche Schnittstellen zur Verwendung der Messsensoren.

5 Funktionale Anforderungen

5.1 GUI

F010 Die Benutzer erreichen nach Öffnung der Anwendung direkt die GUI.

F020 Der Benutzer öffnet durch den Optionen-Knopf die Einstellungen.

F030 Der Benutzer kann durch den Datei-Knopf die Dateiverwaltung öffnen.

F040 Der Benutzer kann durch den Hilfe-Knopf das Hilfe Fenster öffnen.

[1] `https://github.com/GuenterQuast/PhyPiDAQ`

5.1.1 Menüfeld

F050 Der Benutzer öffnet durch den Sensoren-Knopf die Auswahl an Sensoren.

F060 Der Benutzer öffnet durch den Verbindungen-Knopf die Auswahl an Verbindungen.

F070 Der Benutzer öffnet durch den Darstellungen-Knopf die Auswahl an Darstellungen.

F080 Der Benutzer erhält durch verschiedene Farben der Konfigurationsbausteine eine visuelle Repräsentationen ihrer Komplexität.

5.1.2 Optional: Zusätzliche Funktionen im Menüfeld

F090 Der Benutzer kann durch den Sensoren hinzufügen-Knopf weitere Sensoren hinzufügen.

F100 Der Benutzer kann durch den Transformationen hinzufügen-Button weitere Transformationen hinzufügen.

F110 Die in Transformationen in F100 sollen in eigenen Python-Skripten geschrieben werden.

5.1.3 Konfigurationsfeld

F120 Durch Drag und Drop kann der Benutzer Konfigurationsbausteine im Konfigurationsfeld platzieren.

F130 Durch Betätigen des Messung starten-Knopf startet der Nutzer eine Messung.

5.1.4 Darstellungsfenster

F140 Das Darstellungsfenster ist bei der Initialisierung der Anwendung leer.

F150 Durch die Benutzerkonfiguration im Konfigurationsfeld wird durch F090 automatisch die Darstellungsart im Darstellungsfenster geöffnet.

5.2 Konfigurationserstellung

F160 Über F030 kann der Benutzer gespeicherte Standartkonfigurationen öffnen oder alte Messwerte verwenden.

F170 Durch F160 geladene Konfigurationen und Messdaten werden automatisch nach Format überprüft.

F180 Aus F050 kann der Benutzer Sensoren durch Drag und Drop in das Konfigurationsfeld ziehen.

F190 Die Anwendung sollte automatisch überprüfen, ob der ausgewählte Sensor richtig angeschlossen ist.

F200 Die Anwendung gibt durch F190 automatisch eine visuelle Rückmeldung über die Farbe des Sensors und einer Pop-Up Benachrichtigung.

F210 Aus F060 kann der Benutzer eine oder mehrere Verbindungen durch Drag und Drop in das Konfigurationsfeld ziehen.

F220 Die ausgewählten Verknüpfungen kann der Benutzer mit einem ausgewählten Sensor verknüpfen.

F230 Die Anwendung sollte automatisch überprüfen, ob die Verbindungen eine geeignete Kombination mit dem Sensor darstellt.

F240 Aus F070 kann der Benutzer eine Darstellungsart per Drag und Drop in das Konfigurationsfeld zu ziehen.

F250 Die Ausgewählte Darstellung kann der Benutzer mit einer Verbindung verknüpfen.

F260 Die Anwendung sollte automatisch überprüfen, ob die gewählte Darstellungsart eine sinnvolle Repräsentation der Messdaten darstellt.

F270 Über F030 kann der Benutzer seine eigene Konfiguration über einen Konfiguration speichern-Knopf speichern.

F280 Bausteine die ein Benutzer nicht mehr in dem Konfigurationsfeld haben möchte können durch Drag und Drop in die Menüleiste wieder versteckt werden.

5.2.1 Optional: Weiterentwicklung der Konfigurationserstellung

F290 Die Anwendung besitzt einen Check-Knopf, welcher die Benutzerkonfiguriation auf Vollständigkeit und Richtigkeit kontrolliert

F300 Durch F290 wird durch visuelle Darstellung der Konfigurations dargestellt, ob sie verwendet werden kann.

5.3 Messablauf

F310 Der Nutzer kann in den Optionen, die Messlänge und die Wertebereiche festlegen.

F320 Der Nutzer kann durch F130 die Messung nach der Benutzerkonfiguration starten.

F330 Durch F320 wird automatisch mit der visuellen Darstellung der Messdaten begonnen.

F340 Der Nutzer kann durch den Messung Messung löschen-Knopf eine durchgeführte Messung pausieren und die visuelle Darstellung auf den Ausgangszustand bringen.

F350 Durch F340 wird nicht die Benutzerkonfiguration gelöscht.

F360 Durch F340 muss der Benutzer erst die Messung starten um weiter zu messen.

F370 Der Benutzer kann durch den Messung pausieren-Knopf die Messung pausieren.

F380 Der Benutzer kann durch den Messung-fortsetzen-Knopf die Messung fortsetzen.

F390 Der Benutzer kann eine Messung durch F380 nur fortsetzen, wenn sie zuvor durch F370 pausiert wurde.

F400 Der Benutzer kann durch den Messdaten speichern-Knopf die gemessenen Daten speichern.

F410 Der Benutzer kann durch den Graph speichern-Knopf den durch die Messung erzeugten Graphen speichern.

F420 Bei F400 und F410 öffnet sich automatisch das Verzeichnis und der Nutzer muss einen eigenen Dateinamen eingeben und speichern.

5.4 Fehlermeldungen

F430 Durch F170 geladene Konfigurationen und Messdaten werden automatisch nach Format überprüft und eine aussagekräftige Fehlermeldung zurückgegeben.

F440 Durch F240 und F270 wird bei einer ungültigen Kombination von zwei Konfigurationsbausteinen eine aussagekräftige Fehlermeldung zurückgegeben

F450 Die Anwendung sollte dem Benutzer automatisch durch Pop-Up Benachrichtigung darauf hinweisen, dass bei Betätigen des Messung löschen-Button oder bei Schließen der Anwendung die Messdaten ohne vorheriges Betätigen des Messdaten speichern-Button verloren gehen.

F460 Die Anwendung sollte dem Benutzer eine aussagekräftige Fehlermeldung zurückgeben, falls es zu einem Datenabbruch der Messdaten kommt.

5.5 Bedienungshilfen

F470 Bei enthaltenen Konfigurationsbausteine sollen einen Information-Knopf besitzen, über welche der Benutzer Informationen und Hilfestellung bereitgestellt bekommt.

F480 Über den Hilfe-Knopf erhält der Nutzer eine kurze Beschreibung zur Funktionalität der Anwendung.

5.6 Sprache

F490 Die Anwendung ist in deutscher Sprache.

5.6.1 Optional: Internationalisierung

F500 Die Anwendung stellt dem Benutzer weitere Sprachpakete für die GUI zur Verfügung.

F510 Durch F020 ist die Sprache für den Nutzer änderbar.

5.7 Sonstiges

F520 Die in der Anwendung enthaltenen Farben sollten mit Rücksicht auf Benutzer mit Rot-Grün Schwäche oder Farbenblindheit ein barrierefreies Verwendung der Anwendung ermöglichen.

5.7.1 Optional: Sonstiges

F530 Durch F020 sollte der Benutzer die in der Anwendung verwendeten Farben ändern können.

F540 Durch F020 sollte der Benutzer die in der Anwendung verwenden Buchstabengröße verändern können.

6 Produktdaten

Zu speichern sind ausschließlich:

- Konfigurationsdaten
- Physikalische Messdaten

Es sollen keine benutzer- oder personenbezogenen Daten gespeichert werden.

7 Nichtfunktionale Anforderungen

7.1 Produktleistungen

NF010 Auslesen von Sensordaten innerhalb von (?) ms. //TODO PhyPiDAQ Abhängigkeit

NF015 Alle (5?) ms können neue Daten angefordert werden.

NF020 Verarbeitung ausgelesener Daten innerhalb von 10 ms.

NF030 Bis zu drei Sensoren können zeitgleich verwendet werden.

NF040 Bis zu (6?) Transformationen können hintereinander verbunden werden.

NF050 Bis zu (3?) Senken können zeitgleich verwendet werden.

7.2 Benutzbarkeit

NF110 Die Benutzeroberfläche ist für Siebtklässler verständlich.

NF115 Nicht intuitiv verständliche Funktionen werden dem Benutzer erklärt.

NF120 Ungespeicherte Daten werden nicht ohne Warnung verworfen.

NF130 Programmfehler werden dem Benutzer gegenüber klar kommuniziert.

NF140 Schriftgröße von Text kann verändert werden.

NF150 Farbschema von farbigen UI-Elementen ist veränderbar.

NF160 Die Sprache der Benutzeroberfläche kann verändert werden.

7.3 Zuverlässigkeit

NF210 Die Anwendung reagiert konsistent auf Eingaben.

NF220 Korrekte Bedienung und Eingaben führen exakt zu den erwarteten Effekten.

NF230 Unerwartete und fehlerhafte Eingaben führen nicht zum Systemabsturz.

NF240 Verbindungsverlust von Sensoren führt nicht zum Systemabsturz.

7.4 Sonstige

NF310 Die Software ist vollständig quelloffen und frei.[2]

NF320 Die Software ist einfach erweiterbar durch Außenstehende.

NF400 Die Datenschutz-Grundverordnung wird nicht verletzt.

[2] `https://www.gnu.org/philosophy/free-sw.de.html`

8 Globale Testfälle und Testszenarien

T010 Erstmaliges Starten und Anpassen der Anwendung

Testziel Teste das Verhalten der Anwendung beim ersten Aufruf und das Ausführen der Demo. Die Demo nutzt Daten aus einer Datei und benötigt keinen angeschlossenen Raspberry Pi.

Vorbedingung Die Anwendung ist installiert. Zu Sehen ist das offene Fenster des Desktops mit der Verknüpfung zur Anwendung.

1. **Aktion** Der Benutzer startet die Anwendung über die Verknüpfung.

   Reaktion Die Anwendung wird gestartet. Das Hauptfenster öffnet sich. Das Start-Infofenster öffnet sich.

2. **Aktion** Der Benutzer setzt den Haken bei „dieses Fenster zukünftig nicht mehr anzeigen“Knopf und drückt dann auf den „Schließen“Knopf des Start-Infofenster.

   Reaktion Das Start-Infofenster schließt sich. Die Start-Einstellung wird gespeichert.

3. **Aktion** Der Benutzer drückt auf den „Hilfe“-Knopf in der Systemleiste.

   Reaktion Das Fenster zur Hilfe öffnet sich.

4. **Aktion** Der Benutzer drückt auf den „Schließen“-Knopf des Hilfefensters.

   Reaktion Das Fenster zur Hilft schließt sich.

5. **Aktion** Der Benutzer drückt auf den „Laden“-Knopf in der Systemleiste.

   Reaktion Das Fenster zum Laden einer Konfiguration öffnet sich. Dabei wird der voreingestellte Defaultpfad geöffnet.

6. **Aktion** Der Benutzer wählt die Demo-Datei in dem Defaultpfad aus und drückt dann auf den „Laden“-Knopf.

   Reaktion Die Anwendung lädt die Demo-Datei. Die geladene Demo-Konfiguration wird auf der Konfigurationsfläche angezeigt.

7. **Aktion** Der Lehrer drückt auf den „Erstellen“-Knopf.

Reaktion Die Anwendung prüft, ob eine gültige Konfiguration vorliegt. Es liegt eine gültige Konfiguration vor. Die Anwendung zeigt den Konfigurationsstatus „OK“ an.

8. **Aktion** Der Lehrer drückt auf den „Starten“-Knopf.

   Reaktion Die Anwendung verarbeitet die Daten der „Sensoren“ und stellt die Ergebnisse dar.

9. **Aktion** Der Benutzer drückt auf den „Stoppen“-Knopf.

   Reaktion Die Ausführung der Messung wird angehalten. Die Darstellungen zeigen ihren Datensatz zum Zeitpunkt des Stoppens weiter an.

10. **Aktion** Der Benutzer drückt auf den auf das „X“-Symbol zum Schließen der Anwendung.

    Reaktion Es öffnet sich ein Fenster und weist den Benutzer darauf hin, dass nicht gespeicherte Daten und Veränderungen verloren gehen.

11. **Aktion** Der Benutzer drückt auf den „Beenden“-Knopf.

    Reaktion Die Anwendung schließt sich.

    Nachbedingung Die Anwendung ist beendet. Es ist kein Fenster ist mehr geöffnet. Es wird der Desktop angezeigt.

    Ergebnis to do

    Abgedeckte Funktionale Anforderungen: to do

T020 Lehrer erstellt eine eigene Konfiguration für den Unterricht

Testziel Teste eine typische Verwendung der Software anhand des gegebenen Szenarios. Dabei wird das Erstellen, Verändern und Speichern einer Konfiguration getestet.

Vorbedingung Die Anwendung ist gestartet und es liegt keine Konfiguration vor. Die Anwendung hat Zugriff auf einen laufenden Raspberry Pi mit PhyPiDAQ. Die Sensoren die verwendet werden sollen sind ordnungsgemäß angeschlossen.

1. **Aktion** Der Lehrer zieht Sensor S1 in die Konfigurationsfläche.

Reaktion Die Anwendung prüft, ob Sensor S1 angeschlossen ist. Danach wird das Icon von Sensor S1 an der gewählten Stelle erstellt. Dabei werden Defaultwerte eingestellt.

2. **Aktion** Der Lehrer ruft die Optionen des Sensors auf(Rechtsklick Icon, dann Optionen).

   Reaktion Die Anwendung ruft das Fenster für die Optionen von Sensor S1 auf.

3. **Aktion** Der Lehrer setzt Werte im gültigen Bereich ein und drückt auf den „Speichern“-Knopf.

   Reaktion Die Einstellungen werden gespeichert und das Optionsfenster wird geschlossen.

4. **Aktion** Der Lehrer zieht die Transformation T in die Konfigurationsfläche.

   Reaktion Das Icon der Transformation T wird in der Konfigurationsfläche erstellt. Die Werte werden auf default eingestellt.

5. **Aktion** Der Lehrer ruft die Optionen von Transformation T auf.

   Reaktion Das Optionsfenster von Transformation T öffnet sich.

6. **Aktion** Der Lehrer stellt die Werte der Transformation im gültigen Bereich ein und drückt auf den „Speichern“-Knopf.

   Reaktion Die Werte werden gespeichert und das Optionsfenster wird geschlossen.

7. **Aktion** Der Lehrer drückt auf den „Optionen“-Knopf der Konfiguration.

   Reaktion Das Optionsfenster für die Konfiguration öffnet sich.

8. **Aktion** Der Lehrer gibt verändert die Defaultwerte der Konfiguration und drückt auf den „Speichern“-Knopf.

   Reaktion Die Werte werden gespeichert und das Optionsfenster schließt sich.

9. **Aktion** Der Lehrer drückt auf den „Konfiguration Speichern“-Knopf

   Reaktion Die Anwendung öffnet das Fenster zum speichern einer Konfiguration.

10. **Aktion** Der Lehrer wählt einen gültigen Pfad, gibt einen gültigen Namen ein und drückt danach auf den „Speichern“-Knopf.

    Reaktion Die Anwendung erstellt eine Konfiguration-Datei mit entsprechendem Namen am entsprechenden Ort. Das Fenster zum Speichern einer Konfiguration wird geschlossen.

    Nachbedingung Die Anwendung ist geöffnet. In der Konfigurationsfläche ist Sensor S1 und Transformation T zu sehen.

    Ergebnis Diese Konfiguration ist als Datei an dem gewählten Ort gespeichert.

    Abgedeckte Funktionale Anforderungen: to do

T030 Schüler der siebten Klasse bearbeitet eine Aufgabe

Testziel Es soll getestet werden wie ein Schüler der siebten Klasse mit der Anwendung umgeht, während er eine Aufgabe des Lehrer bearbeitet. Dabei ist schon die Konfiguration aus T020 geladen. Diese enthält Sensor S1 und Transformation T.

Vorbedingung Die Anwendung ist gestartet und es liegt eine vom Lehrer erstellte Konfiguration vor. Die Konfiguration enthält Sensor S1 und Transformation A. Die Anwendung hat Zugriff auf einen laufenden Raspberry Pi Pi mit PhyPiDAQ. Die Sensoren S1 und S2 sind ordnungsgemäß angeschlossen.

1. **Aktion** Der Schüler drückt auf das „?“-Icon neben Sensor S1.

   Reaktion Es öffnet sich ein Infofenster zum Sensor S1.

2. **Aktion** Der Schüler drückt auf das „?“-Icon neben Transformation A.

   Reaktion to do: schließt sich Infofenster zu Sensor S1 automatisch oder bleibt es offen bis Schüler es schließt ?

3. **Aktion** Der Schüler drückt auf den „Start“-Knopf.

   Reaktion Ein Fenster öffnet sich und teilt dem Schüler mit, dass eine Konfiguration zunächst erstellt werden muss.

4. **Aktion** Der Schüler drückt auf den „OK“-Knopf.

   Reaktion Das Infofenster zum starten ohne erstellen wird geschlossen.

5. **Aktion** Der Schüler drückt auf den „Erstellen“-Knopf.

   Reaktion Die Anwendung prüft, ob die vorliegende Konfiguration gültig ist. Sie ist ungültig. Es wird ein Fenster geöffnet mit der Meldung, dass keine gültige Konfiguration vorliegt. Es verlinkt zum Hilfemenü.

6. **Aktion** Der Schüler drückt auf die Verlinkung zum Hilfsmenü.

   Reaktion Das Fenster zum Hilfsmenü öffnet sich. (TO DO wird Fehlermeldung geschlossen?)

7. **Aktion** Der Schüler drückt auf den „X“-Knopf des Hilfsmenüfensters.

   Reaktion Das Hilfsmenüfenster wird geschlossen.

8. **Aktion** Der Schüler zieht die Darstellung D in die Konfigurationsfläche.

   Reaktion Ein Icon der Darstellung D wird erstellt. Es werden Defaultwerte geladen.

9. **Aktion** Der Schüler drückt auf das „?“-Icon der Darstellung.

   Reaktion Das Infofenster zu Darstellung D öffnet sich.

10. **Aktion** Der Schüler schließt das Infofenster.

    Reaktion Das Infofenster zu Darstellung D wird geschlossen.

11. **Aktion** Der Schüler drückt auf den Kanal von Darstellung D und danach auf den linken Kanal von Transformation T.

    Reaktion Eine Verbindung zwischen den beiden Kanälen wird erstellt.

12. **Aktion** Der Schüler drückt auf den oberen rechten Kanal von Transformation T und danach auf den Kanal von Sensor S1.

    Reaktion Eine Verbindung zwischen den beiden Kanälen wird erstellt.

13. **Aktion** Der Schüler zieht Sensor S2 in die Konfigurationsfläche.

    Reaktion Das Icon zu Sensor S2 wird erstellt.

14. **Aktion** Der Schüler ruft die Optionen von Sensor S2 auf.

Reaktion Das Optionsfenster von Sensor S2 wird geöffnet.

15. **Aktion** Der Schüler gibt die vom Lehrer vorgegebenen Werte ein und drückt auf den „Speichern“-Knopf.

    Reaktion Die Werte werden gespeichert und das Optionsfenster von Sensor S2 wird geschlossen.

16. **Aktion** Der Schüler drückt auf den Kanal von S2 und danach auf den letzten offenen Kanal von Transformation T.

    Reaktion Eine Verbindung zwischen Sensors S2 und Transformation T wird erstellt.

17. **Aktion** Der Schüler drückt auf den „Erstellen“-Knopf.

    Reaktion Die Anwendung prüft, ob eine gültige Konfiguration vorliegt. Es liege eine gültige vor. Die Anwendung zeigt den Konfigurationsstatus „OK“ an. Dabei wird durch die Voreinstellung der Lehrers aus T020 Aktion 8 und 9 der Messzeitraum auf 10 Sekunden gesetzt.

18. **Aktion** Der Schüler drückt auf „Starten“.

    Reaktion Die Anwendung holt für 10 Sekunden Daten aus den beiden Sensoren, transformiert diese und stellt sie in Darstellung D dar. Nach der Messung stoppt die Anwendung automatisch. Die Darstellung D bleibt erhalten.

19. **Aktion** Der Schüler drückt auf den Knopf „Speichern“ bei der Darstellung.

    Reaktion Ein Fenster zum Speichern des Graphen öffnet sich.

20. **Aktion** Der Schüler wählt einen Pfad und gibt einen Namen ein und drückt dann auf den „Speichern“-Knopf.

    Reaktion Der Graph wird am entsprechenden Ort gespeichert und das Fenster zum Speichern wird geschlossen.

21. **Aktion** Der Schüler drückt auf den Knopf „Messwerte speichern“.

    Reaktion Ein Fenster öffnet sich zum Speichern der Messwerte.

22. **Aktion** Der Schüler wählt die Darstellungsform „Tabelle“, den Pfad, den Namen und drückt auf den „Speichern“-Knopf.

Reaktion Die Messwerte werden als Tabelle mit entsprechendem Namen am entsprechenden Ort gespeichert. Das Fenster zum Speichern schließt sich.

23. **Aktion** Der Schüler drückt auf den „Speichern“-Knopf der Konfiguration.

    Reaktion Die Anwendung öffnet das Fenster zum Speichern einer Konfiguration.

24. **Aktion** Der Schüler wählt dort Name und Pfad aus und drückt auf den „Speichern“-Knopf.

    Reaktion Die Konfiguration wird entsprechend gespeichert. Das Fenster zum Speichern einer Konfiguration wird geschlossen.

    Nachbedingung Die Anwendung ist geöffnet. Nur das Hauptfenster ist offen. In der Konfigurationsfläche ist die vollständige Konfiguration zu sehen. Die Messung wurde abgeschlossen und die Darstellungen sind zu sehen.

    Ergebnis Die Konfiguration, die Messwerte sowie der Graph der Messwerte sind als Dateien gespeichert.

    Abgedeckte Funktionale Anforderungen: TO DO

T040 Umgang mit destruktivem Benutzer

Testziel Teste den Umgang mit einem Benutzer, der versucht alle Grenzen der Anwendung und alle Fehlermeldungen zu finden.

Vorbedingung Geöffnete Anwendung ohne angeschlossenem Raspberry Pi und damit mit keinem gültigen Sensor.

1. **Aktion** Der Benutzer zieht Sensor S3 in die Konfigurationsfläche hinein.

   Reaktion Die Anwendung prüft ob der Sensor S3 angeschlossen ist. Er ist es nicht. Die Anwendung erstellt kein Icon, sondern öffnet eine Fehlermeldung, dass Sensor S3 nicht gefunden wurde.

2. **Aktion** Der Benutzer schließt die Meldung. Danach öffnet er über die Systemleiste das Menü um eine Datei als Sensor zu Initialisieren.

   Reaktion Die Fehlermeldung wird geschlossen. Das Menü zum Initialisieren einer Datei als Sensor wird geöffnet.

3. **Aktion** Der Benutzer wählt eine Datei mit ungültigem Format aus und drückt auf den „Öffnen“Knopf.

   Reaktion Die Anwendung prüft ob die Datei ein gültiges Format hat. Sie hat es nicht. Die Anwendung zeigt eine Fehlermeldung an, dass die gewählte Datei ein ungültiges Format hat.

4. **Aktion** Der Benutzer schließt die Fehlermeldung und wählt eine gültige Datei.

   Reaktion Die Anwendung prüft das Format, erkennt es als gültig und erstellt den Sensor S91. Das Fenster zum laden einer Datei wird geschlossen.

5. **Aktion** Der Benutzer lädt eine weitere gültige Datei als Sensor S92.

   Reaktion Die Anwendung öffnet das Ladefenster, prüft das Format der gewählten Datei, erkennt sie als gültig an und erstellt Sensor S92. Dann wird das Fenster zum Laden einer Datei geschlossen.

6. **Aktion** Der Benutzer zieht erst Sensor S91 und dann Sensor S92 in die Konfigurationsfläche.

   Reaktion Die Anwendung prüft, ob Sensor S91 und S92 angeschlossen sind. Das sind sie, da sie aus gültigen Dateien stammen. Dann erstellt die Anwendung die entsprechenden Icons an den entsprechenden Stellen.

7. **Aktion** Der Benutzer drückt auf den Kanal von S91 und danach auf den Kanal von S92.

   Reaktion Die Anwendung erkennt, dass beide Kanäle von Sensoren stammen. Sie erstellt keine Verbindung und zeigt die Fehlermeldung an, dass zwischen zwei Sensoren keine Verbindung erstellt werden kann.

8. **Aktion** Der Benutzer schließt die Fehlermeldung und zieht Transformation T1 in die Konfigurationsfläche. Dabei hat T1 zwei Eingangskanäle und einen Ausgangskanal.

   Reaktion Die Anwendung schließt die Fehlermeldung. Danach erstellt sie das Icon von T1 an der entsprechenden Stelle mit Defaultwerten.

9. **Aktion** Der Benutzer klickt auf den oberen Eingangskanal und danach auf den unteren Eingangskanal.

Reaktion Die Anwendung erkennt, dass zwei Eingangskanäle verbunden werden sollen. Sie verbindet sie nicht, sondern zeigt die Fehlermeldung an, dass zwei Eingangskanäle einer Transformation nicht direkt verbunden werden können.

10. **Aktion** Der Benutzer schließt die Fehlermeldung. Danach klick er auf den oberen Eingangskanal von T1 und danach auf den Ausgangskanal von T1.

    Reaktion Das Fenster der Fehlermeldung schließt sich. Die Anwendung erkennt, dass die Verbindung ungültig ist und gibt eine Fehlermeldung aus.

11. **Aktion** Der Benutzer schließt die Fehlermeldung. Danach zieht er Transformation T2, Darstellung D1 und Darstellung D2 in die Konstruktionsfläche.

    Reaktion Das Fenster der Fehlermeldung wird geschlossen. Die Icons von T2, D1 und D2 werden an den entsprechenden Stellen erstellt.

12. **Aktion** Der Benutzer klickt auf den Eingangskanal von D1 und danach auf den Eingangskanal von D2.

    Reaktion Die Anwendung erkennt, dass die Verbindung ungültig ist und gibt die passende Fehlermeldung aus.

13. **Aktion** Der Benutzer schließt die Fehlermeldung. Danach schließt er zwei Sensoren S1 und S2 ordnungsgemäß an. Weiter zieht er erst S1 und dann S2 in die Konfigurationsfläche.

    Reaktion Das Fenster der Fehlermeldung wird geschlossen. Die Anwendung erkennt, dass S1 angeschlossen ist und erstellt das entsprechende Icon. Weiter erkennt sie, dass der Sensor S2 zwar angeschlossen ist, aber die maximale Anzahl an Sensoren in der Konfigurationsfläche überschreitet. Die Anwendung erstellt dann kein Icon von S2 und gibt eine passende Fehlermeldung aus.

14. **Aktion** Der Benutzer schließt die Fehlermeldung und zieht Darstellung D3 und D4 in die Konfigurationsfläche.

    Reaktion Die Fehlermeldung wird geschlossen. Die Anwendung erstellt ein Icon für D3 und erkennt, dass mit D4 die maximale Anzahl an Sensoren überschritten wird. Also gibt sie eine entsprechende Fehlermeldung aus.

15. **Aktion** Der Benutzer schließt die Fehlermeldung. Danach verbindet er alle offenen Kanäle ordnungsgemäß und drückt auf den „Erstellen“-Knopf.

Reaktion Das Fenster der Fehlermeldung wird geschlossen. Die Anwendung erstellt alle gültigen Verbindungen. Weiter prüft sie beim Erstellen die Konfiguration und erkennt dessen Korrektheit.

16. **Aktion** Der Benutzer drückt auf den „Starten“-Knopf und wartet 5 Sekunden. Danach löst er die Verbindung von S1.

    Reaktion Die Anwendung führt die Messung durch. Nach dem Kappen von S1 stürzt die Anwendung nicht ab. Allerdings stoppt sie die Messung und gibt eine entsprechende Fehlermeldung aus.

. **Aktion**

Reaktion to do: weitere Testfälle

. **Aktion** maximale Anzahl Transformationen?

Reaktion

. **Aktion**

Reaktion

Nachbedingung TO DO

Ergebnis TO DO

Abgedeckte Funktionale Anforderungen: TO DO

9 Systemmodelle

Das projektierte Gesamtsystem wird aller Voraussicht nach drei Systemprozesse benötigen, die wie in Abbildung 1 dargestellt, miteinander Daten austauschen.

Einer dieser drei Systemprozesse hostet die Java-Virtual-Machine, auf der die zu erstellende Hauptkomponente läuft. Diese enthält die grafische Benutzungsoberfläche, die im Wesentlichen die Erstellung der Werte verarbeitenden Messkonfiguration ermöglicht, und die auch deren Betrieb darstellen kann.

Zum Ansteuern und Auslesen der Sensoren auf dem Rasperry Pi wird das auf Python basierende Framework PhyPiDAQ bereitgestellt. Dieses unterstützt bereits eine ganze

Reihe von Sensoren zur Messung diverser physikalischer Größen der Mechanik, Thermodynamik und Elektrodynamik. Da der Python-Interpreter im Allgemeinen nicht im selben Systemprozess wie die Java-Virtual-Machine ausgeführt werden kann, wird für diesen ein zweiter Systemprozess benötigt.

Um Resourcenknappheit auf dem Rasperry Pi aus dem Weg zu gehen, aber auch zur Vereinfachung von Entwicklung und Demonstration, soll die Hauptkomponente nicht zwingend ebenso auf dem Rasperry Pi ausgeführt werden müssen.

Falls denn nun die Hauptkomponente tatsächlich auf einem anderen Rechner ausgeführt werden sollte, so muss sie mit dem Python-Interpreter hardware- bzw. plattformübergreifend Daten austauschen können. Bisher stellt PhyPiDAQ hierfür jedoch keine Funktionalität zur Verfügung.

Zur Realisierung der hardwareübergreifenden Kommunikation sind die folgenden zwei Lösungsvarianten denkbar:

- PhyPiDAQ wird um eine Kommunikationsschnittstelle erweitert, beispielsweise zur Kommunikation über eine Netzwerkschnittstelle wie z. B. WLAN oder zur Kommunikation über eine Peripherieschnittstelle wie z. b. USB.

- In einem dritten Systemprozess, der auf dem selben Rechner wie PhyPiDAQ läuft, stellt eine Utility-Softwarekomponente einen entsprechenden Kommunikationsdienst bereit. In diesem Fall sollte keine Notwendigkeit bestehen, dass PhyPiDAQ wesentlich erweitert werden muss, und der Datenaustausch mit PhyPiDAQ sollte auf möglichst einfache Art und Weise, beispielsweise per Standardeingabe und Standardausösungsvariante erfolgen können.

Da bei letzterer Variante die Utility-Softwarekompomente zusätzlich auch noch eine Überwachungsfunktion bezüglich PhyPiDAQ übernehmen und dieses im Falle von Abstürzen und Nicht-Responsivität erneut ausführen kann, ist letztere Lösungsvariante die bevorzugte Lösungsvariante zur Realisierung der optionalen hardwareübergreifenden Kommunikation.

Abbildung 1 stellt die Verteilung der Systemkomponenten auf zwei Rechnern mittels der gewählten zweiten Lösungsvariante dar, und benennt die notwendige Utility-Komponente mit "Measurement-Server".

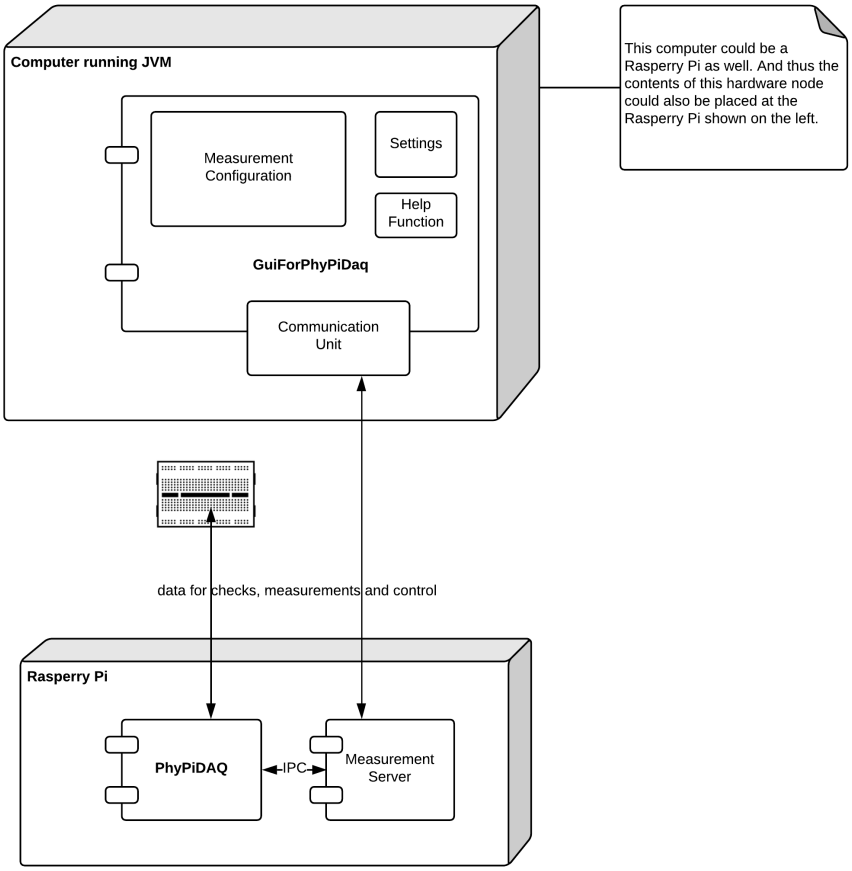

Abbildung 1: UML-Deployment-Diagramm

10 Benutzungsoberfläche

10.1 Ziel der Benutzeroberfläche

Das Ziel der Benutzeroberfläche ist es dem Anwender eine intuitive Benutzung des Programmes zu ermöglichen. Da das Programm im Schulbetrieb eingesetzt werden soll ist eine gute Verständlichkeit wichtig. So soll auch eine lange Einarbeitungszeit für den Anwender vermieden werden. Dabei soll keine Funktionalität verloren gehen.

10.2 Generell

Um die Ziele zu erfüllen wird das Programm über eine Grafische Benutzeroberfläche (kurz "GUI") bedient. Der Anwender soll in der Lage sein bereits vorhandene Computer und Mobile-Device Kenntnisse zu nutzen.

10.3 Eingabegeäte

Die GUI ist überwiegend für die Bedienung durch die Maus ausgelegt. Dabei werden überwiegend folgende Aktionen verwendet:

- Click
- Drag-and-Drop

Eine Tastatur wird benötigt um weitergehende Einstellungen, wie etwa das anpassen einer Option, zu ermöglichen.

10.4 Überblick

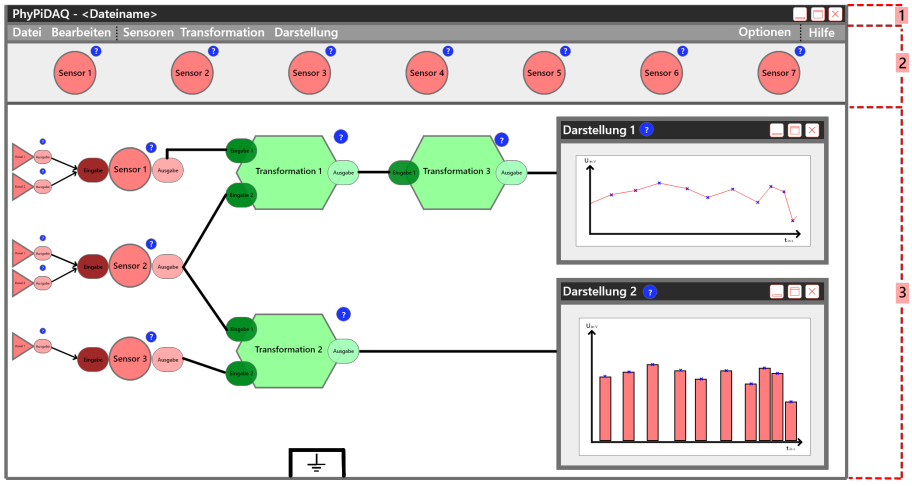

Abbildung 2: Der Grundlegende Aufbau der Hauptbenutzeroberfläche

Abbildung 2 zeigt den Aufbau der Grafischen Benutzeroberfläche. Dabei geben die Zahlen in den roten Kästchen, rechts von dem Programmfenster, die logische Unterteilung an.

Die zu dem jeweiligen Teil gehörenden Elemente sollen im folgenden erklärt werden.

10.5 Die einzelnen Teile

10.5.1 Systemmenüleiste

Der Name des Programmes wird hier angezeigt. Rechts daneben steht, sofern vorhanden, der Name der Datei, die gerade geöffnet ist.

Das "Maximieren"-Symbol vergrößert das Programmfenster auf die maximale Größe. Die Größe ist von der Benutzungsumgebung abhängig.

Das "Minimieren"-Symbol blendet das Programmfenster aus. Es ist weiterhin geöffnet und wieder aufrufbar.

Das "Schliessen"-Symbol beendet die Anwendung. Vor dem Beenden findet eine Abfrage statt, ob der Anwender eventuell vorgenommene Änderungen speichern möchte.

10.5.2 Auswahl

Unter dem Reiter "Datei" finden sich Optionen, die sich auf Dateien beziehen. Die wichtigsten sind:

- Das anlegen einer neuen Datei
- Das Speichern der aktuellen Datei
- Das Öffnen einer bereits erstellten Datei

Unter dem Reiter "Bearbeiten" finden sich Optionen, die das Ändern von Inhalten der aktuell geöffneten Datei ermöglichen. Die wichtigsten sind:

- Das Kopieren eines ausgewählten Objektes
- Das Einfügen eines gespeicherten Objektes
- Das Anpassen eines ausgewälten Elements. Diese Einstellungsmöglichkeiten sind von dem Objekt abhängig.

ToDo

ToDo

ToDO

ToDO

10.5.3 Konfigurationsfeld

Das "Informations"-Symbol zeigt nach einem Click weiterführende Informationen zu dem Element an, zu dem es gehört. So würde beispielsweise bei einer Transformation die Funktion angezeigt werde, die sie realisiert.

Repräsentation eines Sensors, der in PhyPiDAQ erkannt werden kann. Vorraussetzung hierfür ist die Existenz einer Konfigurationsdatei.

ToDO

ToDO

Repräsentation einer Transformation. Abhängig von der Anzahl an Eingängen der Transformation besitzt die GUI dementsprechend viele Eingänge. Jedem Eingang können durch Einfügen von einer Verbindung Daten übergegen werden.

ToDO

ToDO

ToDO

ToDO

10.6 Erweiterungsmöglichkeiten

11 Spezielle Anforderungen an die Entwicklungsumgebung

12 Zeit- und Ressourcenplanung

12.1 Projektphasen

Phase	Verantwortlicher	Zeitraum	Kolloquium
Pflichtenheft	Jan Küblbeck	KW 20–22	04.06.2019
Entwurf	Leon Huck	KW 23–26	02.07.2019
Implementierung	Stefan Geretschläger	KW 27–29, 31	(?)
Klausurenphase	—	KW 30, 32	—
Qualitätssicherung	David Gawron	KW 33–35	03.09.2019
Abnahme	—	KW 36	—
Abschlussprüfung	Linus Ruhnke	KW 37/38	(?)

12.2 Entwurfsphase

12.3 Implementierungsphase

13 Ergänzungen

14 Glossar

Click Der linke Maustasten Click.

Drag-and-Drop Methode um mit graphischen Benutzeroberflächen zu interagieren. Dabei wird ein Objekt erst mit der Maus festgehalt und an einen anderen Ort gezogen. Durch das lösen der Maustaste wird das Objekt platziert.

Konfigurationsdatei Bezeichnet eine Datei, die es PhyPiDAQ erlaubt einen Sensor anzusprechen. Sie beinhaltet Python Code, der die Datenentgegennahme von dem Sensor verwaltet. Für jeden Sensor gibt es eine eigene Konfigurationsdatei. Konfigurationsdateien können angepasst werden um das Messverhalten anzupassen.

Open Source Projekt Ein Open Source Projekt ist ein Projekt, wessen Quelltext und Dokumentationen von dritten angesehen, geändert und genutzt werden kann..

OSL 2 OSL 2 ist der Name eines Open-Source-Lehrsoftware-Labor, in welchem Studierende sich mit der Entwicklung von Open-Source-Software vertraut machen können..

PhyPiDAQ Siehe Abschnitt 4.3 „PhyPiDAQ“.

Raspberry Pi Der Raspberry Pi ist ein Einplatinencomputer. In diesem Projekt dient der Raspberry Pi als Hardwareplattform, um unterschiedliche Sensoren zu verbinden und ihre Daten auszulesen..

Science Labs Ein Science Lab ist ein Arbeitsplatz, welcher Schülern ermöglicht wissenschaftliche Forschungen unter kontrollierten Bedingungen durchzuführen..

Sensor Erfasst unter anderem physikalische Größen. Damit stellt ein Sensor den Grundbaustein für die Messtechnik da.

Transformation Berechnung von neuen Daten aus einem Datenstrom. Die Transformation können durch mathematische Funktionen realisiert werden. Je nach Transformation werden unterschiedlich viele Datenströme als Eingabe erwartet.