Bachelorarbeit Pitch

Daniel Pollack

Inhaltsverzeichnis

1 Grundlegende Ideen

Die Bachelorarbeit soll zum Ziel haben ein Programm zu erarbeiten, mit dessen Hilfe ein informatisch unversierter Nutzer Messungen in einer elektrotechnischen Schaltung vornehmen kann ohne vorhandene Programmierkenntnisse.
Der Titel könnte wie folgt lauten: *"Realisierung eines Datenloggers und Datenerfassungssystems mit Arduino UNO/Mega...*
... zum Einsatz in wissenschaftlicher Experimentierumgebung
..."

2 Grundlagen / theoretischer Hintergrund / Motivation

- Was ist ein DatenLogger:

  Generelle Literatur zur Definition und Klärung wichtiger Terminologie

- Wie sieht die praktische Anwendung aus:

  Feld/Labor

  Kernkomponenten

  Campell Logger, und andere

- Grundlagen Arduino

  Warum, Was, Trend

  Specs

  ggf. Mikrocontroller allgemein

- Case-Study

  Anwendung, Einsatz und Zweck von Datenloggern im Wissenschaftsbetrieb

  Erläuterung von beispielhaften Szenarien

  Welcher DatenLogger soll als Blaupause dienen?; CR100?

  Wie sieht die Programmierung aus (CRBasic vs. EdLog)

  Nachteile / Vorteile gegenüber Arduino

- Spezifikationen Arduino:

  Vorteile und Nachteile gegenüber DatenLogger

  Einschränkungen im Einsatz (Bit-Resolution, Anzahl der Kanäle, nur Single-Ended,...)

  12-Bit A/D-Wandler ist Standard

  16-Bit A/D von Adafruit ist möglich

3 Hauptinhalte

- Nutzerfreundliches Interface
- “einfaches” Handhabung der konfiguration

  Ardoino UNO und/oder MEGA sind zwingend zu unterstützen
- Echtzeitloggen der mehrerer Messwerte
- Laufzeit (ausgenommen laufende Messungen) Eingabe von Übertragungsfunktionen -¿ siehe LUA oder irgendeine Skriptsprache
- OpenSource (vermutlich MIT oder GNU Lizens)
- Konfigurierbarkeit der Diagramme

  Nutzer kann die diagramm darstellung weitestgehen beeinflussen:

  eine beliebige anzahl von diagrammen soll unterstützt werden

  scalierung der Diagramm soll möglich sein

4 Optionale Inhalte

- Interface bietet Möglichkeit zum Aufspielen der Arduino-Software
- Unterstützung mehrerer Chipsätze (Arduino Uno — Arduino MEGA etc.)
- Diagramme können in separate Fenster ausgelagert werden
- Einfache “Reaktionen” sollen umgesetzt werden können

  z.B. Input A geht von **LOW** zu **HIGH** dann setze Output B

5 Software

Die Software soll sich in ihrere Handhabung und ihrem Erscheinungsbild an Nicht-Informatiker richten. Die Konfiguration des Messaufbaus soll über eine grafische Oberfläche ünterstützt werden, so das der Nutzer zu jeder Zeit weis welcher Pin des Arduinos, zum Messzeitpunkt, konfiguriert sein wird.

Für die Log-Dateien soll eine Lokalisierung möglich sein. So soll der Nutzer zum Beispiel wählen können ob er lieber eine englische oder deutsche Zahlendarstellen haben will. Dies ist besonders von Interessen wenn es um die Weiterverarbeitung der Messwerte geht und diese in andere Programme importiert werden sollen.

Das Interface selbst soll nicht lokalisiert werden. Englisch als Sprache ist ausreichend.

6 Arduino

Die Arduino Software soll prinzipiell keine vim Nutzer definierte Logik ausfürhen. Ihre Aufgabe wird es sein auf die, über die bestehende USB-Verbindung empfangenen Befehle zu erkennen und zu verarbeiten. Diese Befehle können in dem Teil Befehlssatz eingesehen werden.

6.1 Befehlssatz

Befehl	Parameter	Richtung	Beschreibung
setDPin	Id des digital Pins	IN	Konfiguriert einen digital Pins als entweder Input oder
setAPin	Id des analog Pins	IN	Konfiguriert einen analogen Pins als entweder Input ode
fetchDPin	Id des digital Pins	OUT	Gibt den aktuellen Zustand eines digital Pins wieder
fetchAPin	Id des analog Pins	OUT	Gibt den aktuellen Wert eines analogen Pins an
readyArduino	-	IN	Prüft ob Arduino fertig initialisiert ist und Befehle entg
readyPC	-	OUT	Antwort auf *readyArduino*. Bestätigt das der Arduino ei
whoAreYou	-	IN	Fragt Arduino nach seinem Model
iAm	ModelId	OUT	Antwort auf *whoAreYou*

6.1.1 Begriffsklärung

IN von PC zu Arduino

OUT von PC zu Arduino mit Rückgabe von Werten an PC

7 Alternativen—Research

7.1 Instrumentino und Controlino

Istrumentino und die dazu gehörige Controlino Software für den Arduino wurden von Israel Joel Koenka, Jorge Sáiz, Peter C. Hauser entwickelt. Die Software verfolg den Ansatz einer Nutzer getriebenen Anpassung der Umgebung an eine bestimmte Aufgabe, wie z.B. die Messung von Werten aus einem Mass-FlowControler. Hierzu wird ein durchaus überdurchschnittliches Informatikfachverständnis vorraus gesetzt.

7.1.1 Vorteile

- eine Konfiguration kann als eigenständiges Programm auf Basis der Bibliothek gesehen werden und ist somit, weitest gehend, vor versehentlicher Änderung der Konfiguration geschütz

7.1.2 Nachteile

- Der Installationsprozess gestaltet sich als umständlich und setzt zu weilen Nieschensoftware vorraus, die zusätzlich installiert werden muss.
- Eine Dokumentation ist nicht vorhanden -¿ Die Konfiguration eines Versuchsaufbaus gestaltet sich als umständlich
- Die Konfiguration der Software kann nur durch eine Person mit Informatik/Python Kenntnissen durchgeführt werden.
- Die Conrtolino Software verfügt über keine mitgelieferte Möglichkeit ohne größere Anpassung des Codes und schreiben eines Makefiles o.ä. für andere Arduino Boards aufgearbeitet und aufgespielt zu werden.
- Die Software selber leidet unter Bugs, die z.B. Grafiken nicht richtig oder garnicht laden

- Die Darstellungen mehrerer Signale wird nur gebündelt in einem Diagramm unterstützt. Es fehlt eine Option zur Erstellung mehrerer Diagramme
- Der CSV Logger beginnt bereits zum Start der Software mit der Logprozess, auch wenn garkein Arduino angeschlossen ist. -¿ Die Logdateien müssen aufgearbeitet und die Leerenzeilen entfernt werden.

7.2 CmdMessenger

Die CmdMessenger Bibliothek basiert auf der Sprache Mono, einer freien Implementation der .net und C# Sprache. Die Bibliothek bietet ein Grundgerüst für einen Arduino-Sketch, welche selbstständig mit Funktionen gefühlt werden muss. Die Anwendungen müssen jedoch von Fall zu Fall entwickelt werden.

7.2.1 Vorteile

- Weitreichende Konfigurationsmöglichkeiten
- gute Dokumentation

7.2.2 Nachteile

- Setzt Programmierkentnisse für sowohl den Arduino, als auch C# vorraus

7.3 Campell Logger

7.3.1 Vorteile

- Umfassende Konfiguration druch direkt Programmierung

7.3.2 Nachteile

- Eigens entwickelte Sprache => muss neu erlernt werden
- Kein Interface für alles.

  Stattdessen ist die Software in Unterprogramme aufgeteilt, die nur bedingt mit einander Kommunizieren
- Ist kostenintensiv in der Anschaffung

  Periverie ist mit zusätzlichen Kosten verbunden