

Universal data acquisition

FS17 Praxismodul

Dane Wicki

15. Juni 2017

Inhaltsverzeichnis

1	Einleitung, Problembeschreibung	2
1.1	Geschäftsfeld der Firma	2
1.2	Projektkontext	2
1.3	Projektziele	2
2	Projektergebnisse	3
2.1	Ergebnisse	3
2.2	Anforderungen	3
3	Umsetzung	4
3.1	Verwendete Tools	4
3.1.1	LabView	4
3.2	Software	5
3.2.1	Bestehende Software	5
3.2.2	Systemgrenzen	6
3.2.3	Komponente UDA	6
3.2.4	Data Preparation & Recorder	7
3.3	Datenbank	7
3.3.1	Bestehende Datenbank	7
3.3.2	Datenbank	8
4	Projektergebnis	8
5	Arbeitsjournal	9
6	Fazit	9
7	Bestätigung Arbeitgeber	10
	Glossary	11

1 Einleitung, Problembeschreibung

1.1 Geschäftsfeld der Firma

Die Siemens AG ist ein führender internationaler Technologiekonzern, der seit mehr als 165 Jahren für technische Leistungsfähigkeit, Innovation, Qualität, Zuverlässigkeit und Internationalität steht. Das Unternehmen ist in mehr als 200 Ländern aktiv und zwar schwerpunktmäßig auf den Gebieten Elektrifizierung, Automatisierung und Digitalisierung. Siemens ist weltweit einer der größten Hersteller energieeffizienter ressourcenschonender Technologien. Das Unternehmen ist einer der führenden Anbieter effizienter Energieerzeugungs- und Energieübertragungslösungen, Pionier bei Infrastrukturlösungen sowie bei Automatisierungs-, Antriebs- und Softwarelösungen für die Industrie. Darüber hinaus ist das Unternehmen ein führender Anbieter bildgebender medizinischer Geräte wie Computertomographen und Magnetresonanztomographen sowie in der Labordiagnostik und klinischer IT.¹

1.2 Projektkontext

Die Firma Siemens BT im Bereich fire safety in Zug ist zuständig für die Entwicklung von Brandmeldern. Um die Qualität der Brandmelder zu gewährleisten werden diese unter Zuhilfenahme verschiedener Apparaturen und Testaufbauten getestet. Dies geschieht bei grösseren Aufbauten automatisch und mit konsistenter Aufzeichnung der Daten, welche der Melder und etwaige Referenz-Geräte erzeugen. Oftmals gibt es jedoch den unterschiedlichsten Aufgabestellungen folgende individuelle Laboraufbauten, bei welchen die Aufzeichnung nur unter grossen Anstrengungen der Arbeitenden auswertbare Ergebnisse liefern.

1.3 Projektziele

Das Hauptziel des Projektinhaltes ist es, mögliche Verbesserungen sowie Ansprüche für die vielen kleinen und individuellen Aufbauten zu finden und auszuschöpfen. Das heisst, die an verschiedenen Arbeitsplätzen zu findenden individuellen Aufbauten sowie deren Software Umgebung, sollen durch eine universelle, gemeinsam nutzbare und flexible Datenerfassungslösung ersetzt werden.

Ein weiteres Ziel ist es, dass der Endanwender eine Möglichkeit bekommt in einem vernünftigen Zeitrahmen eine auswertbare Datei zu erhalten.

Eine weitere Herausforderung stellt die Erneuerung von EN und UL-Normen dar, nach welchen alle Produkte und Melder nach veränderten Ansprüche getestet und geprüft werden müssen. Da Siemens BT fire safety bis anhin die zukünftig inkrafttretenden Anforderungen noch nicht in einem optimierten Prozess testen konnte, wurde

¹ Siemens. [Online; 03.04.2017]. URL: <http://www.siemens.com/>.

ein neuer Testaufbau bestellt, bei welchem diese Optimierungen mit zum Zuge kommen. Es soll erreicht werden, dass die zu erstellende Testsoftware, gemäss den zeitlichen Ansprüchen der produktentwicklungs Projekte, rechtzeitig mit dem neuen Testaufbau in Betrieb genommen werden kann.

2 Projektergebnisse

2.1 Ergebnisse

Die folgenden Ergebnisse müssen im Rahmen dieses Projektes erarbeitet werden:

- DB Skripte für die Erstellung der Datenbank
- Endsoftware
- Installationsanleitung
- Bedienungsanleitung
- SW-Dokumentation

2.2 Anforderungen

Die folgenden Punkte muss die Software erfüllen:

1. Name des neuen Programmes ist "Universal data acquisition" UDA
2. Das Programm muss auf Win7, 8.1, ... laufen.
3. Modularer Aufbau
4. Alle Eingaben sollen auf ihre Plausibilität überprüft werden.
5. Es soll in einem gleichbleibenden Intervall aufgezeichnet werden können.
6. Bei falschen und undefinierten "Objekten" sollen Fehlermeldungen mit Angabe der Fehlerquelle aufgelistet werden.
7. Programmeinstellungen sollen in einer ini-Datei gespeichert werden.
8. Die Installation soll mit einem Installer geschehen.
9. Bestehende Funktionalitäten sollen übernommen werden.
10. Die Software muss in LabVIEW geschrieben werden.

3 Umsetzung

3.1 Verwendete Tools

Für die Entwicklung wurden folgenden Komponenten verwendet:

- LabVIEW 2014SP1 (Version 14.0.1f3 32bit)
- OpenGDS v1.0.37(32 bit)
- OpenG v4.0.1.9(32 bit)
- MySQL ODBC 5.2.6(32bit)
- MySQL ODBC 5.2.6(64bit)
- MySQL Server 5.7.14
- MySQL Workbench 6.3.6

3.1.1 LabVIEW

LabVIEW ist eine grafische Programmiersprache, welche von National Instruments entwickelt wird. Die Funktionsweise von LabVIEW ist zudem sehr speziell, da es Funktionsblöcke gibt, welche als Virtuelle Instrumente (VIs) bezeichnet werden. Man kann sich die VIs als Funktionen oder Methoden vorstellen. Sie besitzen immer ein Frontpanel sowie ein Blockdiagramm, in welchem auch der Code zu finden ist (Siehe Figure 1).

Zudem unterstützt LabVIEW seit geraumer Zeit Objektorientiertes Programmieren, dies jedoch nur sehr eingeschränkt verwendbar.

Die mangelnde Fähigkeit zur Objektorientierter Programmierung führt zu dem Umstand, dass an manchen Orten mit einer anderen Art und Weise angegangen werden muss, wie man es sich von Objektorientierten Programmieren gewohnt ist. Zudem sind LabVIEW Programme nicht sonderlich schnell, weshalb besonders stark auf die Performance zu achten ist.

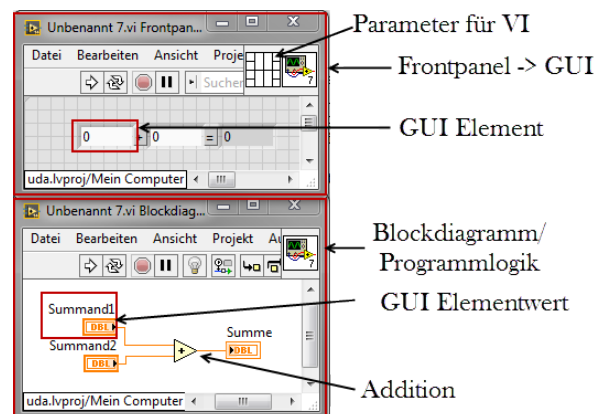


Figure 1: Beispiel LabViewVI mit Beschreibung

Eine weitere Besonderheit ist die Verknüpfung von GUIs mit dem Programm. Es besteht eine sehr starke Verknüpfung zwischen der Funktion und dem GUI. Dies bietet einige Vor- wie auch Nachteile. So wird in diesem Projekt stark mit Subpanels gearbeitet, welches die Möglichkeit liefert dynamisch GUIs von anderen VIs in das laufende Programm einzubinden.²

3.2 Software

3.2.1 Bestehende Software

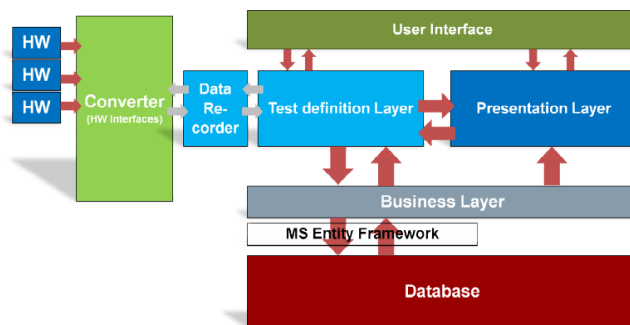


Figure 2: Systemview des Fire Test Commander

Im Rahmen des Umzuges der Testabteilung wurden die Brandräume neu gebaut. Während des Baus dieser Brandräume wurde zudem die veraltete Software, welche für die alten Brandräume erstellt wurde, durch eine Neue ersetzt. Die neu erstellte Software, welche unter dem Namen Fire Test Commander, fortan nur noch FTC genannt, wurde mit LabVIEW erstellt.

Diese Software hat schon

viele Ansätze, welche für die im Rahmen dieses Projektes zu implementierenden Programmes angewandt und übernommen werden können. Der FTC bietet bereits eine Struktur um mit möglichst geringen aufwand Hardwarekomponenten hinzuzufügen (Siehe Figure 2). Diese Struktur, dient zudem gleich als Vorlage für die neu zu entwickelnde Software.

²LabVIEW. [Online; 22.05.2017]. URL: <https://de.wikipedia.org/wiki/LabVIEW>.

3.2.2 Systemgrenzen

Da die Hardware Abstraktion komplett von der bestehenden Software übernommen werden kann, fällt diese aus dem Projekt heraus. Nur die UDA selber, soll als eigenständige Software entwickelt werden. Die Abgrenzung des Systemes kann in folgender Abbildung (Figure 3) nachvollzogen werden. Die Datenbank ist eine MySQL Datenbank, welche mithilfe des ODBC ANSI Treibers angesprochen werden kann. Die Schnittstelle wird von LabVIEW zur Verfügung gestellt und musste auch nicht selber implementiert werden.

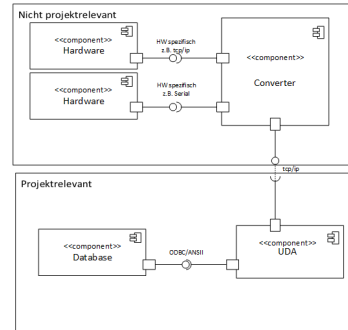


Figure 3: Systemabgrenzung der einzelnen Komponenten

3.2.3 Komponente UDA

Aufgrund von Performance Gründen wurde die Software mit vielen verschiedenen Task ausgestattet. So ist jede GUI Komponente, welche auf dem Hauptprogramm aufrufbar ist, ein eigener Thread, welcher seinen eigenen Event-Handler besitzt. Die GUI Komponenten werden zur entsprechenden Zeit dynamisch ins Hauptprogramm als Subpanels geladen. Diese komplexe Thread Struktur wird im Bild 5 dargestellt.

Diese Software Struktur basiert auf dem Queued Message Handler Template von National Instruments. Das Hauptprogramm (Main) besteht aus zwei parallelen Tasks:

- dem Event-Handler, der primär auf Eingaben vom Anwender reagiert
- und der Verarbeitungstask (Queued Message Handler), der die Aufträge aus seiner Message-Queue entnimmt und der Reihe nach abarbeitet.

Die Bedienoberfläche umfasst vier Subpanel-VIs. Diese werden im Hauptprogramm aufgerufen und laufen parallel zu den beiden Tasks des Hauptprogrammes. Das Frontpanel eines dieser Subpanel-Vis wird programmatisch im Subpanel des Hauptprogrammes sichtbar gemacht. Jedes dieser Subpanel-Vis besitzt einen eigenen Event-Handler der auf Eingaben des Anwenders reagiert. Sowohl der Event-Handler des Hauptprogrammes wie auch diejenigen der Subpanel-Vis senden Messages an die Message-Queue der Verarbeitungstask im Hauptprogramm (In der Abbildung 5 mit grünen durchgezogenen Pfeilen dargestellt).

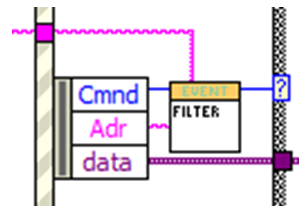


Figure 4: Aufruf des FilterVIs

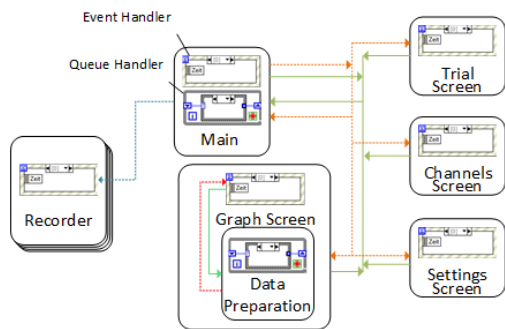


Figure 5: Laufzeitansicht der GUI Tasks und deren Kommunikation

Die im Bild 5 dargestellten Pfeile stellen die Kommunikationswege dar. Es gibt insgesamt 2 Möglichkeiten der Kommunikation. Wie oben geschrieben gibt es die Message-Queue, mit welcher Messages an das Hauptprogramm gesendet werden können. Dieses wiederum kann mit allen anderen über ein User-Event (In der Abbildung 5 mit orangen gestrichelten Pfeilen dargestellt) kommunizieren. Die User-Events werden an alle Event-Handler gesendet (Broadcast). Mit Hilfe eines Filter-Vis (siehe Figure 4) kann der empfangende Event-Handler die Events aber filtern, sodass er nur Events mit der richtigen Adresse

(ID) verarbeitet.³

3.2.4 Data Preparation & Recorder

Die Umsetzung setzte voraus, dass das Aufzeichnen aller Daten in einem immer gleich bleibenden Intervall aufgezeichnet werden. Dazu war nötig, dass für jede angeschlossene Hardware ein eigener Thread startet, welcher die Daten zwischenspeichern kann. Um dies zu gewährleisten wurde ein Recorder-VI erstellt, welcher über einen Asynchronen Aufruf mehrmals gestartet werden kann. Die Kommunikation zu diesen VIs findet vom Hauptprogramm über einen Speziellen Event an alle laufenden Recorder statt (Siehe Figure 6). Dieser Event wird jedoch an keinen der anderen Task gesendet. Der Event lässt schließlich alle Recorder starten. Dieses Verhalten ist mit dem NotifyAll in Java vergleichbar, womit alle Thread zur selben Zeit gestartet werden können.

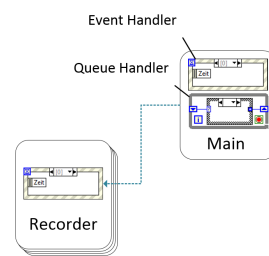


Figure 6: Kommunikation Main mit Recorder

3.3 Datenbank

3.3.1 Bestehende Datenbank

Mit der Implementierung der FTC Software wurde auch eine eigens dafür entwickelte Datenbank aufgebaut. Diese ist an sich enorm Komplex und sehr verstrickt, weshalb

³Rupert Kobras, Paul Wermelinger, and Martin Schnyder. *Code Dokumentation FTC*. Siemens internal Document. Siemens.

hier nicht viel näher auf diese eingegangen wird.

3.3.2 Datenbank

Da, wie schon erwähnt Software übernommen wird, so etwa die gesamte Hardwareabstraktion, muss auch die Datenbank an diesen Stellen übernommen werden. Um die Funktionalität der übernommenen Komponenten zu gewährleisten, wurde ein DB Schema der bisherigen Datenbank so abgespeckt (Siehe Figure 7), dass sie für die UDA brauchbar ist.

Im Figure 7 wurden zusammengehörende Tabellen farblich zusammengelegt. Die Blau markierten Tabellen dienen dem einzelnen Versuch. So werden alle Visuellen Kanäle in der Visual Tabelle abgelegt. Hinter diesem sichtbaren Kanal liegt ein Physikalischer Kanal, welcher in der Physical Tabelle zu finden ist. An diesen Physikalischen Kanal sind die Meldeinformationen, welche allesamt in der Grün markierten Fläche zu finden sind, angeschlossen, sowie die Hardware Kommunikationssinformationen, welche allesamt in der Gelb markierten Fläche zu finden sind, angeschlossen. In allem erhält man so einen Testaufbau, welcher dynamisch aus vordefinierten Geräten sowie Hardware Abstraktionen zusammengesetzt werden kann, in dem man die Blau markierten Tabellen verändert.



Figure 7: Vereinfachtes DB-Schema für die UDA

4 Projektergebnis

Es konnten die wesentliche Komponenten für die Aufzeichnung der Daten, sowie das Anzeigen jener Daten in einer neuen Datenerfassungssoftware erstellt werden. Diese Komponente werden in weiteren Schritten verbessert und erweitert. Die Software wurde schon nach den ersten erfolgreichen Datenaufzeichnungen verwendet und ist im Moment bei 2 Testaufbauten in Betrieb. Sie speichert die Daten persistent und erstellt auswertbare Excel Dateien, mit welchen die Testdesigner Auswertungen machen können. Eine Installationsanleitung erleichtert zudem die Installation, da die Anleitung Schritt für Schritt mit Bildern erklärt wird. Die Softwaredokumentation sowie die Bedienungsanleitung wurden in Absprache mit den Vorgesetzten im Moment noch

klein gehalten, da die Notwendigkeit einer funktionierenden Aufzeichnung die Dokumentation in den Hintergrund drängte. Der Grund für die unübliche Dringlichkeit, der Erstellung einer funktionierenden Aufzeichnungskomponente lag an einer Anpassung einer UL Norm, welche dafür sorgte, dass Melder mit einer neuen Apparatur, ohne vorhandene Datenerfassungssoftware, möglichst frühzeitig im Projekt getestet werden sollte.

5 Arbeitsjournal

Tätigkeit	Zeitraum	Aufwand (d a 8h)
Analyse der Anforderungen	W1	1
Einarbeitung in die Bestehende Software	W1 - W2	4
Planung	W3	2.5
SW-Design	W3-4	1.5
DB-Modell	W4	1.5
DB-Erstellen (Installieren und DB Modell umsetzen)	W4 - W5	1
GUI Design	W5	2
DB Kommunikation	W5 - W6	2.5
Settings Screen (ini file Modifikation)	W6 - W7	2
Channels Screen (Anzeige von DB Info)	W7	1.5
Graph Screen (ohne Kommunikation)	W8 - W9	4
Recorder Implementation (Kommunikation zu Hardware Komponenten)	W9 - W10	3.5
Implementation Excel Export	W10 - W11	3
Graph Screen Inbetriebnahme mit Kommunikation	W11 - W12	3
Fertigstellung der Grundfunktionalität (Zusammenführen der Komponente)	W13	2
Installer erstellen	W13 - W14	1
Inbetriebnahme (Gebrauch der Software mit Ankunft des neuen Aufbaus)	W14	3
Installationsanleitung	W14	1
SW-Dokumentation (Beginn bei Planung wurde jeweils ergänzt und am Schluss nochmals zusammengetragen)	W14	0.5
Summe:		40.5

6 Fazit

Für mich war und ist die UDA ein enorm grosses Projekt, welches mich sicher noch ein weiteres Jahr beschäftigen wird, da es noch viele Features zu implementieren gilt. Ich muss jedoch gestehen, dass ich die Arbeit unterschätzt habe. Es gab ja architektonisch schon eine Software von der ich Komponenten und Teile übernehmen konnte, so stellte mir LabVIEW jedoch sehr viele Hindernisse in den Weg. Bei allen Objekten, welche ich von der bestehenden Software übernehmen wollte, musste ich viele Teile der Software neu implementieren, da LabVIEW viele Probleme mit Abhängigkeiten bereitete. Weiter hatte ich einige Schwierigkeiten, bis ich das Aufzeichnen der Daten hingekriegt habe. Hier half jedoch, dass die Hardwarekomponenten schon klar eine Schnittstelle definiert hatten. Es dauerte jedoch sehr lange bis ich eine passende Lösung gefunden habe, da ich für die Integrität des Lesens, also dass die Software wirklich immer in einem gewissen Zeitintervall die Daten speichert, für jede Hardware Komponente einen eigenen Thread erstellen musste. Denn LabVIEW weist hier auch so seine Eigenheiten auf, welche ich beachten musste. Als Fazit für mich muss ich leider sagen, dass ich persönlich im Moment für Projekte dieser Grössenordnung von LabVIEW abraten würde. Denn obwohl man schnell das Skelett des GUIs und eine gewisse Grundfunktionalität hat, lässt der Mangel an Wartbarkeit die Software Komplexität steigen. LabVIEW unterstützt zwar inzwischen OOP und unit Tests, so sind diese jedoch im Moment noch nicht ausgereift und lassen die Verwendung teilweise nur eingeschränkt zu.

7 Bestätigung Arbeitgeber

Auftraggeber

Firma	Siemens Schweiz AG
Name, Vorname	Schmid, Urs
Funktion	Head of FireLab
Strasse, Hausnummer	Gubelstrasse 22
PLZ, Ort	6300, Zug
Telefon	+41 79 503 9712
Email	urs.schmid@siemens.com

Verfasser

Firma	Siemens Schweiz AG
Name, Vorname	Wicki, Dane
Funktion	Werkstudent(Informatik)
Strasse, Hausnummer	Gubelstrasse 22
PLZ, Ort	6300, Zug
Telefon	+41 79 503 5181
Email	dane.wicki@siemens.com

Urs Schmid, Auftraggeber

Dane Wicki, Verfasser

References

- [1] Rupert Kobras, Paul Wermelinger, and Martin Schnyder. *Code Dokumentation FTC*. Siemens internal Document. Siemens.
- [2] *LabVIEW*. [Online; 22.05.2017]. URL: <https://de.wikipedia.org/wiki/LabVIEW>.
- [3] *Siemens*. [Online; 03.04.2017]. URL: <http://www.siemens.com/>.

Glossary

Blockdiagramm Das Blockdiagramm ist die Logik Komponente eines LabVIEW VIs.

4

Frontpanel Frontpanel ist die GUI Komponente eines LabVIEW VIs. 4, 6

LabVIEW LabVIEW ist ein grafisches Programmiersystem, welche von National Instrument entwickelt wird. 3–6, 10, 11

Subpanel LabVIEW GUI Element mit welchem ein Laufendes VI in dieses Element geladen werden kann.. 5

VI VI, Virtual Instrument. Ist eine Funktion in der LabVIEW Umgebung. 4, 5, 7, 11

Acronyms

FTC Fire test commander. 5

UDA Universal data acquisition. 3