

Universal data acquisition

FS17 Praxismodul

Dane Wicki

May 23, 2017

Inhaltsverzeichnis

1	Einleitung, Problembeschreibung	2
1.1	Geschäftsfeld der Firma	2
1.2	Projektkontext	2
1.3	Problembeschreibung	2
1.4	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
3.3	Datenbank	8
3.3.1	Bestehende Datenbank	8
3.3.2	Datenbank	8
3.4	Software Architektur	9
3.4.1	LabView Tasks	9
3.4.2	Recorder	10
3.4.3	GUI / Tasks	10
3.4.4	Objek	10
3.5	Datenbank	10
3.5.1	Datenbank-Classen	10
3.6	Testing	10
4	Projektergebnisse	10

5 Arbeitsjournal	10
6 Fazit	10
7 Bestätigung Arbeitgeber	10

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.

1.2 Projektkontext

Die Firma Siemens BT 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 vielen Aufbauten automatisch und mit konsistenter Aufzeichnung der Daten, welche der Melder und etwaige Referenz-Geräte erzeugen. Es gibt jedoch weiterhin aufbauten, bei welchen die Aufzeichnung weder Automatisch noch Konsistent gespeichert werden kann oder nur unter grossen Anstrengungen der Arbeitenden. Diesen Zustand gilt es nun zu verbessern. Dazu soll eine Software entwickelt werden, die aus verschiedenen Ressourcen (verschiedenen Datenquellen) die Daten sammelt und diese in eine auswertbare Excel-Datei exportiert. Diese Software basiert auf einer bestehenden Software, welche für das Brandlabor entwickelt wurde. Es sollen dabei Bestandteile der dieser Bestehenden Software verwendet werden. Es sind also folgende Ziele zu erfüllen:

1.3 Problembeschreibung

1.4 Projektziele

Siemens Building Technologies ist unter anderem ein Produkthersteller im Bereich Brandschutz. Dazu gehört die Herstellung verschiedener Brandmelder. Um sich der Qualität jener Brandmelder gewiss sein zu können, werden diese in verschiedenen eigens erstellten Testanlagen getestet. Sie werden getestet, um auch sicherzustellen

das die Brandmelder bei der jeweiligen Zertifizierungsstelle durchkommen. Bei einem Solchen Testdurchlauf werden alle Daten der jeweiligen Brandmelder aufgezeichnet und anschliessend ausgewertet.

Es gibt eine grosse Variation dieser Testanlagen. Bei den grösseren Testaufbauten wurden eigene Software erstellt, welche auch zur Automation genutzt werden. Es gibt jedoch auch einige etwas kleinere Testaufbauten, bei welcher keine Software vorhanden ist, welche die zu sammelnden Daten aufzeichnet. Es wird momentan bei jeder dieser Anlagen auf eine umständliche Art und Weise getestet. Dies stellt eines der Probleme dar. Dieser Umstand führt auf einen erhöhten Zeitaufwand. Zudem kommt bei manchen Aufbauten dazu, dass sie selten gebraucht werden. Der fehlende Zyklische Gebrauch der jener Aufbauten führt zu einer erhöhten Einarbeitungsperiode. Weiter kommt ein neuer Standard für einen bestehenden Brandmelder hinzu. Dieser neue Standard führte dazu, dass die Testabteilung der Siemens einen neuen Testaufbau bei der Firma XYZ bestellte. Im Rahmen der Bestellung wurde jedoch nur der Aufbau bestellt, keine Passende Software, welche alle Daten während eines Testlaufes aufzeichnen könnte. Für diesen Zweck gilt es eine Software zu entwickeln.

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.

- Name des neuen Programmes ist "Universal data acquisition" UDA
- Das Programm muss auf Win7, 8.1, ... laufen.
- Modularer Aufbau
- Alle angaben sollen auf Ihre Plausibilität überprüft werden.
- Bei Falschen, und undefinierten "Objekten" sollen Fehlermeldungen mit Angabe der Fehlerquelle aufgelistet werden.
- Programmeinstellungen sollen in einer ini-Datei gespeichert werden.

- Die Installation soll mit einem Installer geschehen.
- Bestehende Funktionalitäten sollen übernommen werden.
- Die Software muss in LabView geschrieben werden.

3 Umsetzung

3.1 Verwendete Tools

Für die Entwicklung wurde folgende Software 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 graphische Programiersprache, welche von National Instruments entwickelt wird. Die Funktionsweise von LabVIEW ist zudem sehr speziell es gibt Funktionsblöcke, welche Virtuelle Instrumente (VIs) bezeichnet werden. Diese VIs 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 unzulässig oder sehr eingeschränkt.

Wegen diesem Umstand, der mangelnden Objektorientierter Programmierfähigkeit, musste an einigen orten mit einer anderen Art und Weise angegangen werden, wie man es sich gewohnt ist. Zudem sind LabView Programme nicht sonderlich schnell, weshalb es viele Verschiedene Tasks gibt.

Eine weitere Besonderheit ist die Verknüpfung von GUIs mit dem Programm, es ist sehr stark mit dem GUI verknüpft, welches Vor- wie auch Nachteile bietet. So wird bei unserem Projekt stark mit Subpanels gearbeitet, welches die Möglichkeit liefert dynamisch GUIs von anderen VIs in das laufende Programm einzubinden.

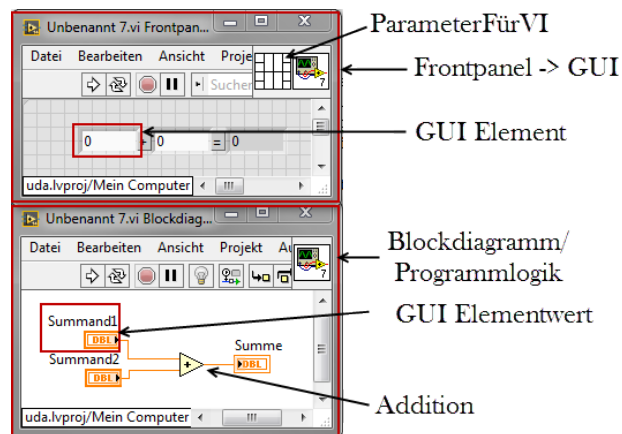


Figure 1: Beispiel LabView VI mit Beschreibung

3.2 Software

3.2.1 Bestehende Software

Im Rahmen eines Umzuges der Testabteilung, wurden die Brandräume neu gebaut. Während des Baus wurde zudem die veraltete Software, welche für die Alten Brandräume erstellt wurde. Die neu erstellte Software, welche unter dem Namen Fire Test Commander fortan nur noch FTC, wurde mit LabView erstellt. Diese Software hat schon viele Ansätze, welche für die im Rahmen dieses Projektes zu implementierendes Programm angewandt und übernommen werden können.

Der FTC bietet schon 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.

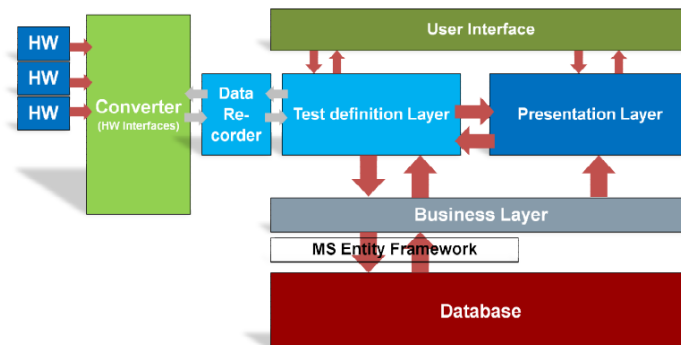


Figure 2: Systemview des Fire Test Commander

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 als eigenständige Software entwickelt. Die Abgrenzung des Systemes, kann in folgender Abbildung (Figure 3) nachvollzogen werden.

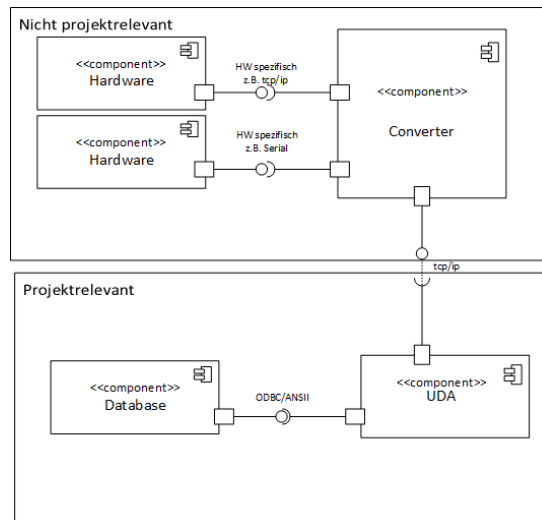


Figure 3: Systemabgrenzung der einzelnen Komponenten

3.2.3 Komponente UDA

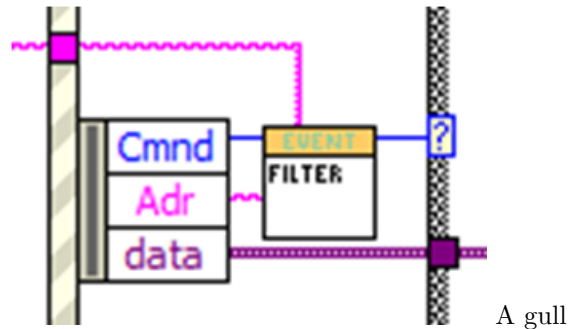
Die Softwarestruktur der FTC-Software basiert auf dem Queued Message Handler Template von National Instruments. Das Hauptprogramm (Main) besteht aus zwei parallelen Tasks

- 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 mit grünen durchgezogenen Pfeilen dargestellt).

Aufgrund der Besonderheiten von LabView wurde die Software mit vielen verschiedenen Task ausgestattet. So ist jede GUI Komponente, welche auf dem Hauptprogramm aufrufbar ist ein eigener Thread. Die Gui Komponenten werden zur entsprechenden Zeit dynamisch in der Hauptkomponente als Subkomponenten geladen. Diese Komplexe Thread Struktur wird im Folgenden Bild dargestellt.

Die Im Bild 4 dargestellten Pfeile stellen die Kommunikationswege dar. Da es nicht nur eine Möglichkeit gibt mit den Threads zu kommunizieren sondern zwei. Die Erste ist mithilfe einer Queue gelöst, welche alle Threads teilen, dabei kann noch eine r0.15



Die Kommunikation vom Queue Handler zu den Event-Handlern der Subpanel-Vis des Hauptprogramms erfolgt mit User-Events (In der Abbildung 4 mit orangen gestrichelten Pfeilen dargestellt). Die User-Events werden an alle Event-Handler gesendet (Broadcast). Mit Hilfe eines Filter-Vis (siehe Figure ??) kann der empfangende Event-Handler die Events aber filtern, sodass er nur Events mit der richtigen Adresse (ID) verarbeitet.

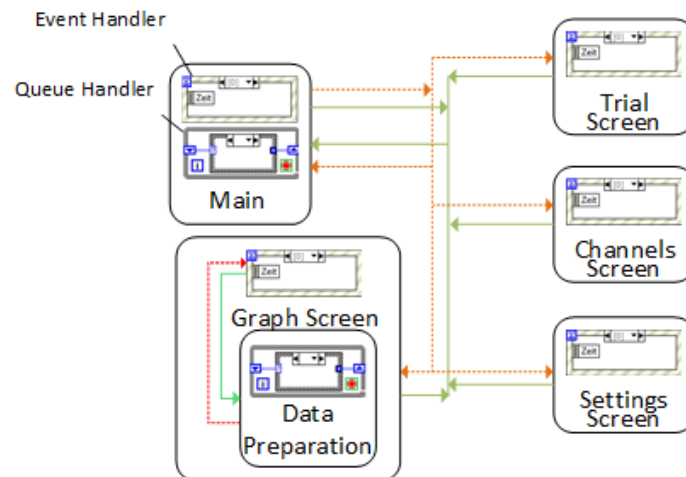


Figure 4: Laufzeitansicht der GUI Tasks und deren Kommunikation

3.3 Datenbank

3.3.1 Bestehende Datenbank

Mit der Implementierung der FTC Software wurde auch eine eigens dafür entwickelte Datenbank aufgebaut. Diese ist in sich enorm komplex und sehr verstrickt, weshalb ich hier nicht viel näher auf diese eingehen kann.

3.3.2 Datenbank

Da wie schon erwähnt Software übernommen wird (So die gesamte Hardwareabstraktion) muss auch die Datenbank an diesen Stellen, mit einigen kleineren Anpassungen, übernommen werden. Um die Funktionalität der übernommenen Komponenten zu gewährleisten wurde eine DB Schema der bisherigen Datenbank so abgespeckt (Siehe Figure 5), dass Sie für die UDA brauchbar ist.

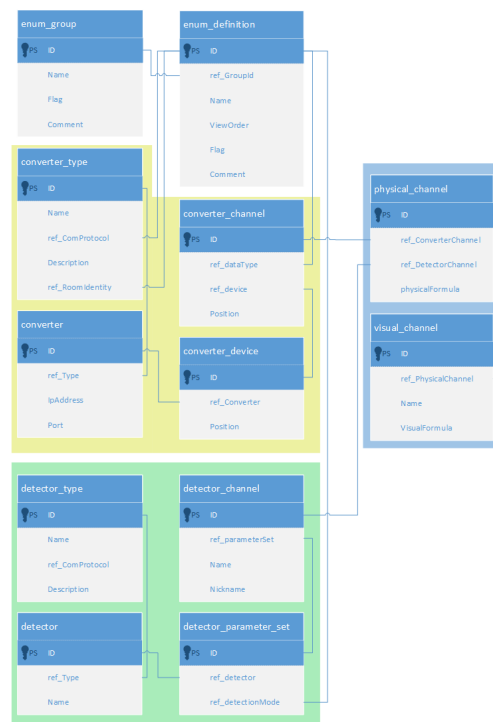


Figure 5: Vereinfachtes DB-Schema für die UDA

Die Bestehende Datenbank des FTC ist sehr komplex und beinhaltet sogar eine Rechteverteilung. Da die neue Software um einiges schlanker wird und auch nicht eine solche komplexe Struktur der Datenbank verlangt, werden nur Teile der bestehenden Datenbank entnommen, welche mit der Hardwareabstraktion zu tun hat. Dies

gewährleistet, dass die Komponente, welche übernommen wird ohne Probleme weiterhin funktioniert.

3.4 Software Architektur

Die System Grobstruktur ist von der bestehenden FTC Applikation übernommen worden und sieht wie folgt aus (siehe Figure 6). Es soll im Vergleich zur FTC Applikation jedoch keine Hardware Abstraktion mehr implementiert werden, da diese übernommen werden kann.

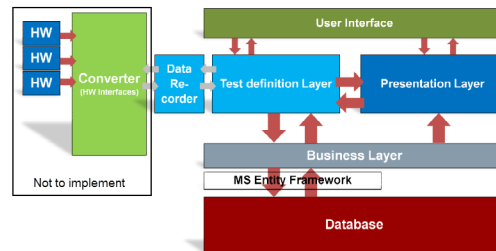


Figure 6: Systemview der UDA

3.4.1 LabView Tasks

LabView

3.4.2 Recorder

3.4.3 GUI / Tasks

3.4.4 Objek

3.5 Datenbank

3.5.1 Datenbank-Classen

3.6 Testing

4 Projektergebnisse

5 Arbeitsjournal

6 Fazit

7 Bestätigung Arbeitgeber

References

[Lab,] Labview.