

**QC-Toolkit**

Jerome Bergmann

Lukas Prediger

Außerdem Anwesend: Deni Raco, Evgeni Wachnowezki, Simon Humpohl

**RWTH Aachen****Anfang:** 12:30, **Ende:** 15:30**Nächster Termin:** 03.03.2015, 10:00**Bericht 1****20.02.2015****1.1 Experiment:**

- Quantenzustand eines Qubits wird über einen Spannungsverlauf (Steuersignal) manipuliert
- Auswirkungen des Steuersignalverlaufs auf Quantenzustand soll untersucht werden
- Verschiedene Geräte für Ausgabe des Steuersignals (AWG) und Messung des Quantenzustands vorhanden
  - grundsätzlich voneinander unabhängig
  - hardwareseitig zeitlich synchronisiert (durch sog. Marker)
- Messzeitpunkte hängen vom Steuersignal ab (Steuersignal muss Messzeiträume berücksichtigen und zu diesen einen bestimmten Spannungswert halten/annehmen)
- Steuersignal wird logisch unterteilt in Pulse, kurze Spannungsverläufe die eine bestimmte Veränderung des Quantenzustands bewirken
- In einem Messlauf wird ein Spannungsverlauf mehrfach gemessen, um statistisch tragbare Daten zu erhalten
- Problem: Steuersignal und Messgeräte müssen unabhängig voneinander konfiguriert werden, aber so, dass an sinnvollen Stellen gemessen wird

**1.2 Vorhandene Software:**

- gesplittet in zwei Module: specialmeasure und pulsecontrol
- specialmeasure: Erlaubt die Konfiguration der Messgeräte und die Durchführung von Messläufen (sog. Scanlines)
- pulsecontrol: Erlaubt die Konfiguration von Steuersignalen/Pulsen und den Upload auf AWG-Geräte
- Momentan stellen beide Module Funktionen bereit, die auf structs arbeiten.
- Globale Daten beider Frameworks stehen auch in structs bereits (smdata, plsdata, awgdata)

**1.2.1 Pulse:**

- Werden für Hardware dargestellt als Waveform: Spannungssequenz über Zeit
- specialmeasure erlaubt verschiedene Definitionsmöglichkeiten (Waveform, pulsetab, weitere)
  - pulsetab: Sequenz von Punkten (Zeit,Spannungsamplitude) aus denen dann die Waveform mittels linearer Interpolation generiert wird

- Pulsgruppen sind Sequenzen von Pulsen oder Pulsgruppen und können parametrisierbar sein (z.B. Amplitude, Wartezeiten..)
- Pulse werden persistent in einer Datei abgelegt und haben eine eindeutige ID
- Pulse können verschiedene Ausgabespannungen für verschiedene Kanäle definieren

### Struct Pulsgruppe (pg):

- Name: `pg.name`
- Puls: `pg.pulses` (Identifiziert über ID)
- # Wiederholungen: `pg.nrep`
- Kanäle: `pg.chan` : Mapping von den in Pulsen definierten Kanälen. Z.B.: Muster (initially ex. [1,2]) invertieren [2,1]

Puls und # Wiederholungen können sowohl einzelne Werte, als auch Arrays von werten sein.

### 1.2.2 Schlüsselfunktionen:

- `awgadd(struct pulsgruppe)`: Lädt Pulsgruppe als Waveform auf den AWG
- `smrun()`: Startet eine Scanline

### 1.2.3 Spezifikationsdetails:

- `uint16` wird als Ein- und Ausgabetyt für Hardware verwendet. Intern wird bei PXDAC/Tectronic `uint10/uint14` verwendet.
- Matlab verwendet standardmäßig immer `double`

### 1.2.4 Hardwaretreiber:

- Objektorientierte Umsetzung der Hardwaretreiber momentan in Arbeit
- AWG-Gerätetreiber werden durch abstrakte Klasse AWG abstrahiert
- Klasse VAWG: Abstraktionsebene: Fasst mehrere AWG-Instanzen als einen virtuellen AWG zusammen (zwecks Rückwärtskompatibilität mit aktuellem Code)

## 1.3 Wortschatz

- Waveform ist ein Puls: (Wird generiert (Endvektor))
- Pulstab: Punktweise angabe (Dazwischen wird interpoliert)
- Scanline: Ein Messvorgang (1x SMRun)
- VAWG: virtueller AWG, bestehend aus mehreren AWG, die ein großes Setup darstellen (abstraktionsebene) (AWG)
- PXDAC (PCI-Bus): Gerät lädt modulationen in den RAM und kann nur von REG A bis REG B n mal abspielen
- Measure entspricht einem Puls der ausgemessen wird(hier werden die Werte dann zB gemittelt)