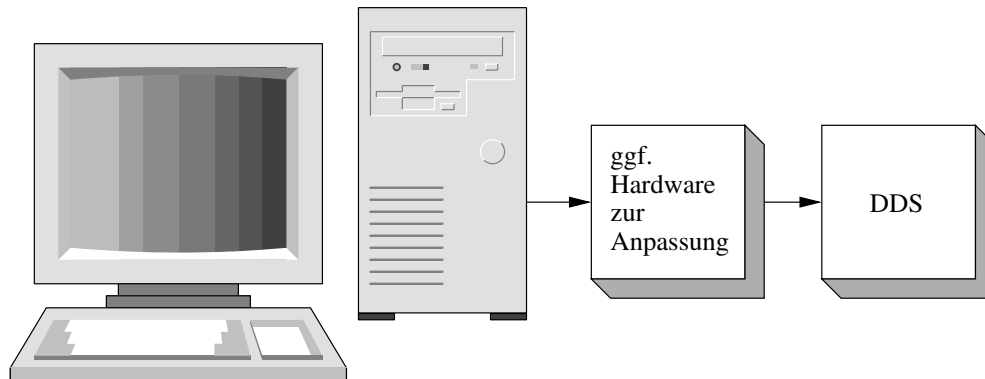


7.11.A Projekte/DDS-Bibliothek – Arbeitsblatt

1 Thema

Es soll in C eine Funktionsbibliothek erstellt werden, mit deren Hilfe ein DDS-Funktionsgenerator (DDS = *direct digital synthesis*) über eine PC-Schnittstelle angesteuert werden kann.



Die Funktionsbibliothek soll so modular sein, dass die gewählte Schnittstelle (seriell, USB per Adapter oder parallel) ohne große Änderung ausgetauscht werden kann.

Ebenso soll es möglich sein, die Funktionsbibliothek auf einen Mikrocontroller zu portieren.

Mit einem kleinen Hilfsprogramm soll die Funktionsbibliothek getestet und vorgeführt werden.

Zum Testen der Software soll die zugehörige Hardware, bestehend aus DDS-Modul nach Applikationsschaltung des Herstellers und ggf. dem Schnittstellen-Adapter, erstellt werden.

2 Rahmenbedingungen Hardware: Mechanik

Die Hardware soll den unmittelbaren Anschluss eines Frequenzzählers oder eines Oszilloskops mit BNC-Eingang ermöglichen. Zur Montage des DDS-Bausteins wird eine Adapter-Platine empfohlen.

3 Rahmenbedingungen Hardware: Elektrik

- Als DDS-Baustein wird der Typ AD9833 empfohlen.
- Bei Benutzung der Parallelschnittstelle soll (wie bei der Applikationsschrift) folgende Belegung verwandt werden: Pin2=D0=SCLK, Pin3=D1=SDATA, Pin4=D2=FSYNC.
- Bei Benutzung der USB-Schnittstelle und eines USB-Seriell-Wandlers soll eine ähnliche Belegung verwendet werden: Bit0=SCLK, Bit1=SDATA, Bit2=FSYNC. Die direkte Verwendung der Handshake-Leitungen zusammen mit einem USB-Seriell-Adapter ist nicht sinnvoll, weil der Erfolg vom Adaptertyp abhängt. Daher soll bei der Benutzung der USB-Schnittstelle ein zusätzlicher seriell-parallel-Wandler verwendet werden (z.B. nach Kainka: MSR mit der RS232-Schnittstelle, S.40ff.).
- Bei direkter Benutzung der RS232-Schnittstelle des PCs können die Handshake-Leitungen zur Ansteuerung verwendet werden, es wird jedoch die gleiche Lösung wie bei USB empfohlen.

4 Rahmenbedingungen Software: API

Die Funktionsbibliothek soll folgende Funktionen zur Verfügung stellen:

```
int dds_set_frequency(unsigned long frequency); /* Frequenz in Hz */
int dds_set_waveform(int waveform); /* 1=SQU, 2=SIN, 3=TRI */
int dds_init(void); /* ggf. Schnittstelle oeffnen */
void dds_exit(void); /* ggf. Schnittstelle schliessen */
void dds_reset(void); /* DDS-Baustein ruecksetzen */
```

5 Rahmenbedingungen Software: Modularität

Die Software soll aus mindestens zwei Schichten bestehen. Die obere Schicht soll die API zur Verfügung stellen. Die untere Schicht soll mit der Hardware kommunizieren. Die obere Schicht soll die Hardware niemals direkt ansprechen, sondern nur über die untere Schicht. Das Testprogramm soll ein eigenes Modul sein, das (außer evtl. zu Testzwecken) nur per API mit der oberen Schicht kommuniziert.

6 Rahmenbedingungen Software: Portabilität

Mit Hilfe bedingter Compilierung soll erreicht werden, dass das Verhalten der unteren hardware-nahen Schicht geändert werden kann, wenn ein anderer PC-Schnittstellen-Typ verwendet wird.

7 Dokumentation

Die Dokumentation der Software soll folgende Punkte enthalten:

- eine allgemeine Beschreibung der Software
- zu jeder Funktion (hier: im Quelltext!) eine kurze Aufstellung gemäß der vorgestellten Vorlage
- Aufrufgraph (*call graph*), d.h. ein Diagramm zur Aufrufhierarchie der Funktionen, evtl. mit Zuordnung der Funktionen zu den Modulen