

Semester/Gruppe/Team:	Abgabedatum:	Protokollführer:
Versuchstag:		weitere Versuchsteilnehmer:
Hochschullehrer:		
Kommentar des Hochschullehrers:		

In die schriftliche Ausarbeitung der Versuche gehört unter anderem:

- VGL, LMN , PRO 7/ 2016

2 Praktikumsaufgabe

Die Aufgabenstellung besteht aus 2 Teilen:

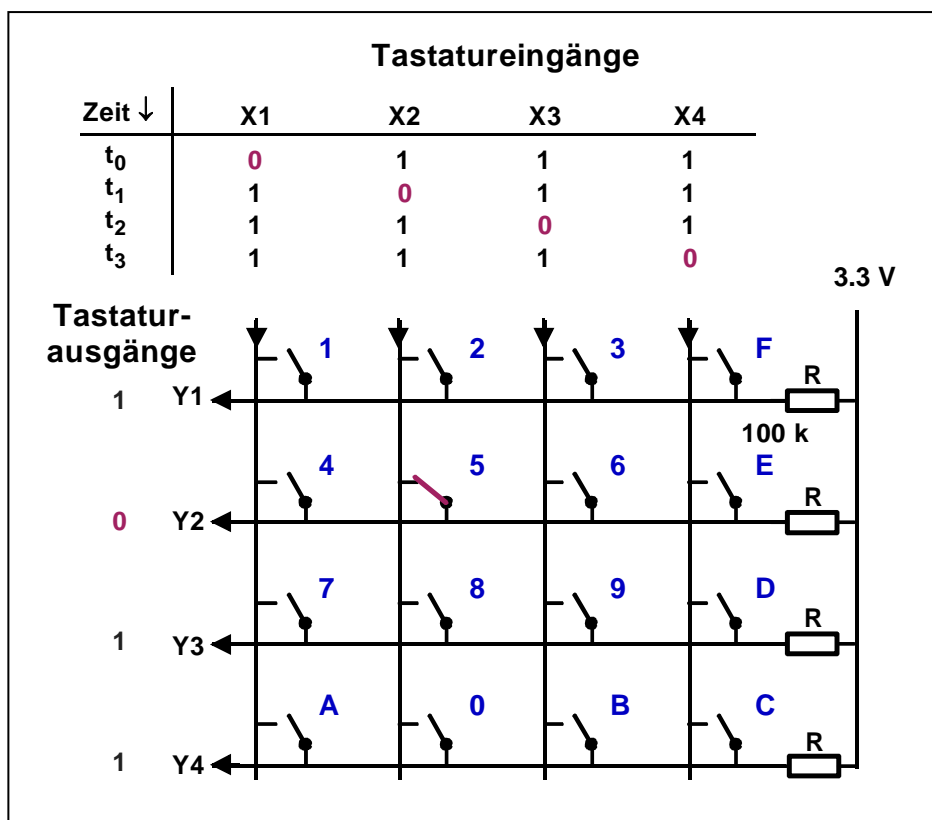
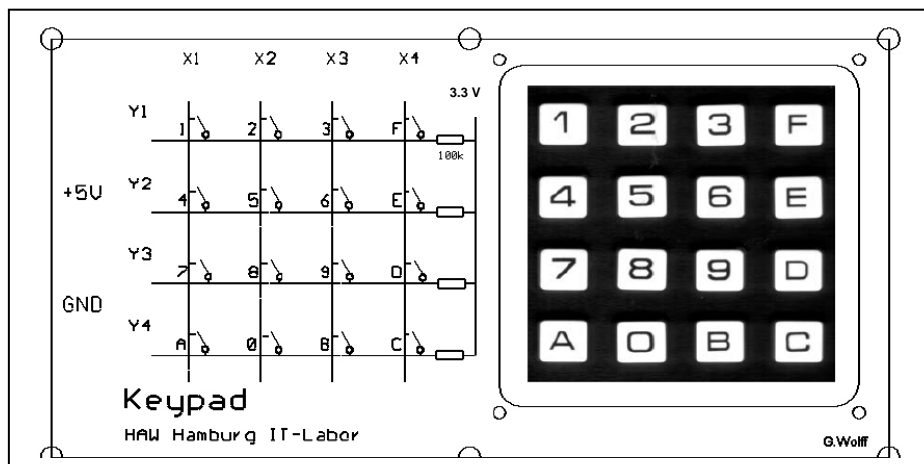
Im 1. Teil sollen Sie Ausführungszeiten und Signallaufzeiten bestimmen.

Im 2. Teil soll eine Tastatureingabe realisiert werden. Dazu steht die Hexadezimal-Tastatur des *MP-Labors* (vgl. *Abbildung 1*) zur Verfügung.

3 Versuchsvorbereitung (vor Beginn des Praktikums zu erledigen!)

3.1 Funktionsweise der Tastatur

Machen Sie sich an Hand von *Abbildung 1* und 2 mit der Funktionsweise der Hexadezimal-Tastatur des *MP-Labors* nach vertraut:



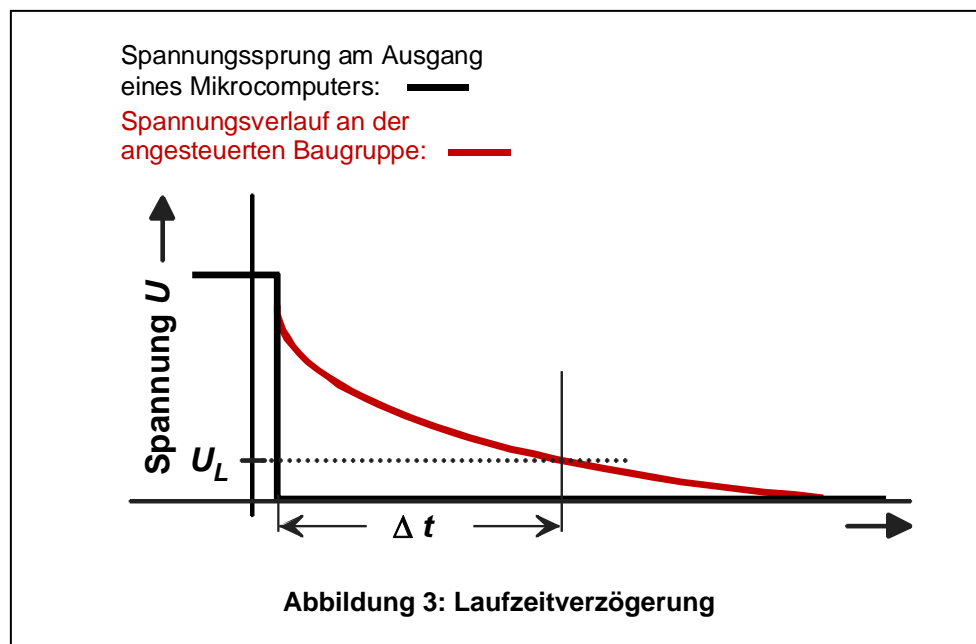
- Die 4 **Eingänge** der Tastatur werden von den **Ausgängen** eines Ports (z.B. Port M) angesteuert.
- Die 4 **Ausgänge** der Tastatur werden mit **Porteingängen** (z.B. von Port M) verbunden.
- Man legt nacheinander jeweils einen der Tastatureingänge auf '0'. Alle Anderen werden auf '1' gesetzt.
- Aus dem Eingang, an dem '0' anliegt und aus dem Ausgang, an dem die '0' erscheint, lässt sich ermitteln, welche Taste gedrückt wurde.

In *Abbildung 2* wurde die Taste „5“ gedrückt, weil zum Zeitpunkt t_1 die '0' von Eingang **X2** am Ausgang **Y2** erscheint.

3.2 Laufzeitverzögerungen in Mikrocomputerschaltungen

Wegen der hohen Verarbeitungsgeschwindigkeit von Mikrocomputern müssen Laufzeitverzögerungen berücksichtigt werden.

Das Beispiel in *Abbildung 3* zeigt einen Spannungssprung von '1' (*High-Pegel*) auf '0' (*Low-Pegel*) an einem Pin eines Mikrocomputerausgangsports (schwarz dargestellt). Dieser Pin steuert eine relativ lange Verbindungsleitung zu einer Baugruppe (z.B. eine Tastatur) an.



Infolge der kapazitiven Belastung im Versuchsaufbau verläuft die Spannung an der Baugruppe entsprechend einer Exponentialfunktion (rot dargestellt), so dass sich der Low-Pegel U_L erst nach einer Zeitverzögerung Δt einstellt. Zusätzliche Signalverzögerungen können durch Baugruppen selber verursacht werden.

Um diese Signallaufzeiten auszugleichen, müssen Mikrocomputerprogrammen häufig die Auswertung so lange verzögern, bis die Pegel ausreichend stabil sind. Hierzu werden Wartefunktionen eingesetzt.

Wartefunktionen werden normalerweise mit einem integrierten Timer realisiert. Solange die Verwendung der Timer nicht bekannt ist, kann eine Warteschleife beispielsweise mit folgender **for**-Schleife in der Funktion **warte** realisiert werden:

```
void warte (unsigned long zeit)
{
    unsigned long j ;
    for ( j = 0 ; j < zeit ; j++ ) ;
}
```

Die Ausführungszeit einer Programmschleife in C (hier realisiert mit: **for (j = 0 ; j < zeit ; j++);**) hängt von der Kompilierung (Umsetzung in Maschinenbefehle) ab und ist im Allgemeinen nicht direkt bekannt. Deshalb muss das Zeitverhalten gemessen werden, um den erforderlichen Wert für den Parameter bei einer vorgegebenen Verzögerung bestimmen zu können.

4 Versuchsdurchführung im MP-Labor

Führen Sie im Labor die nachfolgenden Versuche durch. Erstellen und Verifizieren Sie die dafür notwendigen Programme. Weisen Sie die Funktionstüchtigkeit der Programme nach!

4.1 Ausführungszeitmessung mit dem Oszilloskop

Schreiben Sie ein C-Programm, welches an einem Pin eines Ausgangsports des Mikrocomputers Tiva TM4C1294 eine periodische Rechteckimpulsfolge erzeugt! Verwenden Sie für die Einstellung der Periodendauer die Funktion `void warte (unsigned long zeit)` aus *Unterpunkt 3.1* oder eine vergleichbare selbst geschriebene Funktion! Die Anzahl der Schleifendurchläufe muss nicht vom Terminal eingelesen werden. Sie kann mittels des Debuggens eingestellt werden.

Verwenden Sie ein Oszilloskop, um die Verzögerungszeit für eine unterschiedliche Anzahl von Schleifendurchläufen in der Wartefunktion zu messen! Ermitteln Sie, ob die Periodendauer linear zur Anzahl der Schleifendurchläufe der Wartefunktion verläuft! Erläutern Sie in der Auswertung das Ergebnis!

4.2 Signallaufzeitmessung

Für die Auswertung der Hexadezimal-Tastatur muss das Signal vom Ausgangspin des Mikrocontrollers durch die Tastatur bis zum Eingangspin zurück laufen und benötigt hierfür entsprechend Zeit. Bestimmen Sie mit Hilfe des Rechtecksignals und des Oszilloskops die Laufzeitverzögerung.

4.3 C-Programme zum Einlesen einer Taste der Hexadezimal-Tastatur und Ausgabe auf dem Terminal

Schreiben Sie ein C-Programm, das in einer Endlosschleife feststellt, welche Tasten der Hexadezimal-Tastatur gedrückt sind. Das Ergebnis soll mit `printf(..)` ausgegeben werden!

Anforderungen:

Anforderung 1: Jede Taste soll nur einmal dargestellt werden, solange die Taste gedrückt ist.

Anforderung 2: Werden mehrere Tasten gleichzeitig gedrückt, so soll eine Fehlermeldung ausgegeben werden.

Anforderung 3: Ergänzend zu Anforderung 1, wird eine Taste länger als 1 s gedrückt, so wird diese kontinuierlich hintereinander dargestellt. Die Periodendauer soll 1 s betragen.

Hinweise:

- Beachten Sie bei der Auswertung der Signale der Hexadezimal-Tastatur die Laufzeitverzögerung.
- Implementieren Sie Ihr Programm stufenweise und integrieren die Anforderungen nach und nach.
- Entwerfen Sie Ihr Programm allerdings direkt so, dass es alle Anforderungen umsetzen kann.
- Das Programm soll so konzipiert sein, dass es ggf. in anderen Versuchen wiederverwendet werden kann.