

## Matrixtastatur - Keypad

### 1 Zusatzbaugruppe

Für dieses Praktikum wird eine Zusatzbaugruppe mit einer Matrixtastatur und einer achtstelligen 7-Segmentanzeige an das CT-Board angeschlossen. Der Anschluss erfolgt über P5 und P6 des CT-Boards (siehe Anhang 4.3).

Es ist normalerweise nicht sinnvoll, jede Taste einer Tastatur über eine eigene I/O-Leitung anzusprechen. Um Anschlussleitungen zu sparen, werden die Tasten in einer Matrix-Anordnung aufgebaut.

Die Matrixtastatur wird direkt über die GPIOs des Microcontrollers angesteuert (siehe CT-Board Handbuch Abschnitt 9.2). Die Beschaltung ist in Abbildung 1 ersichtlich. Die Zeilen (PB7..PB4) und Spalten (PB3..PB0) können jeweils separat angesteuert und ausgelesen werden.

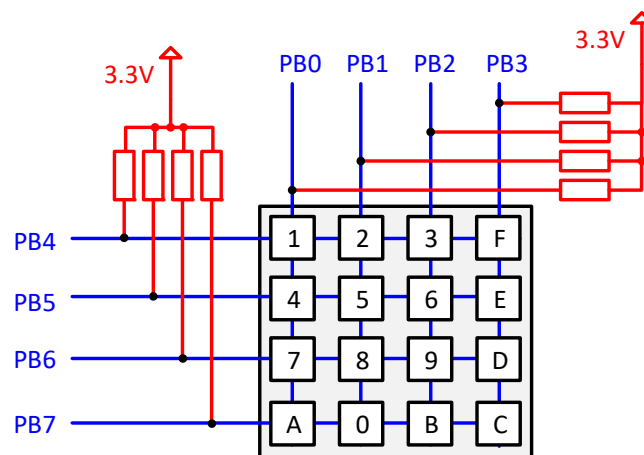


Abbildung 1: Anschluss der Matrixtastatur

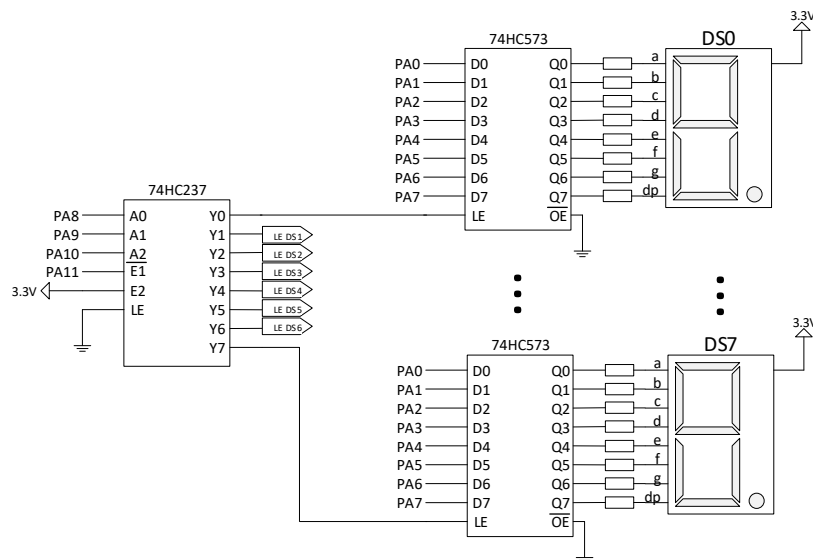
Die Matrixtastatur soll active-low sein. Deshalb sollen die internen Pull-Up Widerstände des Microcontrollers eingeschaltet werden (hier rot markiert).

Die Matrixtastatur kann auf zwei Arten ausgelesen werden.

- 1) **Polling Column by Column:** Es wird genau eine Spalte auf Low gesetzt. Durch Einlesen der Zeilen kann festgestellt werden, ob und welche Taste gedrückt wurde. Dieser Vorgang wird für alle Spalten wiederholt.
- 2) **Fast scanning:** Zunächst werden die Ports der Spalten als Outputs definiert und jene der Zeilen als Input. Nun werden alle Spalten auf Low gesetzt und alle Zeilen eingelesen. Danach kehrt man Outputs und Inputs um und setzt alle Zeilen auf Low. Jetzt können die Spalten eingelesen werden. Kombiniert man nun die Werte von Zeilen und Spalten erhält man ebenfalls die gedrückte Taste.

Der Anschluss der 7-Segmentanzeigen ist in Abbildung 2 ersichtlich. Hierbei handelt es sich um ein Prinzip Bild. Die vollständige Beschaltung ist dem Schema (siehe Anhang 4.4) zu entnehmen. Die acht 7-Segment-Anzeigen verfügen je über ein Latch (74HC573). Die Steuersignale werden gepuffert. Solange der LE-Eingang auf High ist, werden die Daten (D7..D0) an die Ausgänge (Q7..Q0) weitergeleitet. Ist der LE-Eingang auf Low, so werden keine neuen Daten übernommen, der zuletzt gespeicherte Wert liegt an den Ausgängen an.

**Achtung:** Die Anzeigen sind active-low, d.h. sie leuchten, wenn logisch 0 anliegt.



**Abbildung 2: Anschluss der achtstelligen 7-Segmentanzeige**

Die LE-Eingänge der Latches werden über einen Multiplexer, kurz MUX, (74HC237) angesteuert. An den Adressleitungen (A2..A0) kann ein Binärwert angelegt werden, um den entsprechenden Ausgang (Y7..Y0) auszuwählen. Ebenfalls muss der Enable-Eingang ( $\overline{E1}$ , active-low) entsprechend gesetzt sein. Weitere Informationen sind dem Anhang 4.2 zu entnehmen.

Achten Sie darauf, dass sowohl die Adresseingänge auf den Latches als auch die Adresseingänge auf dem Multiplexer stabil sind, wenn Sie das Enable-Signal  $\overline{E1}$  toggeln, d.h. die GPIO Signale für die Adresseingänge und das Enable-Signal dürfen nicht im gleichen Zugriffszyklus geändert werden.

Auf der Zusatzbaugruppe sind zwei DIP-Switches sowie zwei LEDs zu finden. Die Funktionen zur Ansteuerung sind im Modul *inout* gegeben. Die DIP-Switches sollen für das Umschalten zwischen verschiedenen Modi verwendet werden. Die Bedeutung der einzelnen Modi ist im vorgegebenen Hauptprogramm vorgegeben.

## 2 Aufgaben

### 2.1 Ansteuerung der Anzeige

Implementieren Sie die Funktionen `disp_reg_new_value()` und `disp_update()` im Modul `display`. Die Beschreibungen der Funktionen finden Sie im Header File. Die erste Funktion dient dazu, eine Ziffer von rechts in ein Schieberegister (Buffer) einzuschieben. Die zweite zeigt diesen achtstelligen Buffer auf den 7-Segmentanzeigen an.

Testen Sie das Modul `display` indem Sie die zwei vorgegebenen Modi im Hauptprogramm implementieren:

- `MODE_HW_TEST` All digits on
- `MODE_NR_SEQUENCE` The sequence '01234567' shall be displayed

- Port A ist der Ausgabe-Port für die achtstellige 7-Segment-Anzeige.
- Beachten Sie, dass Sie nur die jeweiligen I/Os manipulieren, welche für die Anzeige notwendig sind und die anderen unverändert lassen. Verwenden Sie für die Manipulation der GPIOs das zur Verfügung stehende Modul `hal_gpio`.
- Das Programm ist in folgende Module aufgeteilt:

<code>keypad</code>	Einlesen der Tasten
<code>display</code>	Ausgabe der Werte
<code>inout</code>	Einlesen der Modi-Switches und Ausgabe an LEDs
<code>main.c</code>	Realisiert die Anwendung unter Verwendung der Module <code>keypad</code> , <code>inout</code> und <code>display</code> .

### 2.2 Auslesen der Tastatur: Polling Column by Column

→ `MODE_COL_BY_COL`

Implementieren Sie die Funktion `scan_keypad_cbc()`, welche die Tastatur spaltenweise ausliest und den Wert der gedrückten Taste (0x0 bis 0xF) zurückgibt. Falls keine Taste gedrückt wurde, soll der Wert 0xFF zurückgegeben werden. Die Tastenwerte sollen im Modul `main` auf der achtstelligen 7-Segmentanzeige dargestellt werden. Der neue Wert soll an der Position rechts erscheinen und die vorgängigen Werte sollen um je eine Stelle nach links geschoben werden. Es werden also immer die letzten acht Werte angezeigt.

- Die Matrixtastatur verfügt über eine interne Entprell-Schaltung. Durch diese beeinflussen sich zwei benachbarte Spalten. Bevor also eine neue Spalte auf 0 gesetzt wird, müssen alle Spalten kurzzeitig auf 1 gelegt werden.

### 2.3 Auslesen der Tastatur: Fast scanning

→ `MODE_FAST_SCAN`

Implementieren Sie die Funktion `scan_keypad_fast()`, welche die Tastatur durch Umschalten von Ein- und Ausgängen ausliest und einen entsprechenden Tastenwert zurückgibt. Die Werte der gedrückten Taste sollen wie im Modus 0 über die Anzeige dargestellt werden.

### 3 Bewertung

Das Praktikum wird mit maximal 5 Punkten bewertet:

- Aufgabe 1                      1 Punkt
- Aufgabe 2                      2 Punkte
- Aufgabe 3                      2 Punkte

Punkte werden nur gutgeschrieben, wenn die folgenden Bedingungen erfüllt sind:

- Der Code muss sauber strukturiert und kommentiert sein.
- Das Programm ist softwaretechnisch sauber aufgebaut.
- Die Funktion des Programmes wird erfolgreich vorgeführt.
- Der/die Studierende muss den Code erklären und zugehörige Fragen beantworten können.

## 4 Anhang:

### 4.1 Belegung der GPIOs

Port A:

PA11	PA10	PA9	PA8	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
O	O	O	O	Outputs							
Enable	A2	A1	A0	7-Segment-Data							

Port B:

PB11	PB10	PB9	PB8	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
O	O	I	I	I/O				I/O			
LED2	LED1	S2	S1	Matrix Rows				Matrix Columns			

### 4.2 Datenblattauszug des 74HC237 (Multiplexer)

#### 6. Functional description

Table 3: Function table

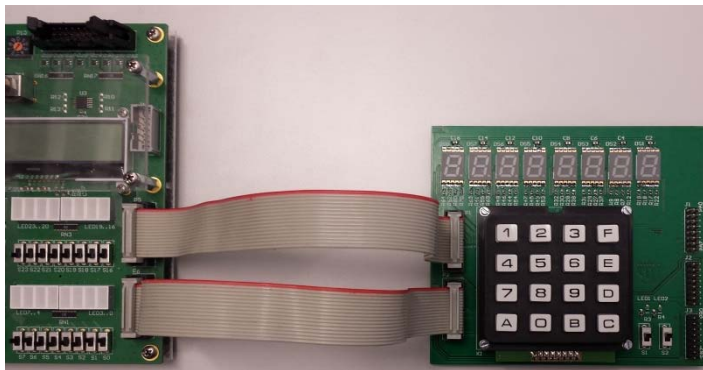
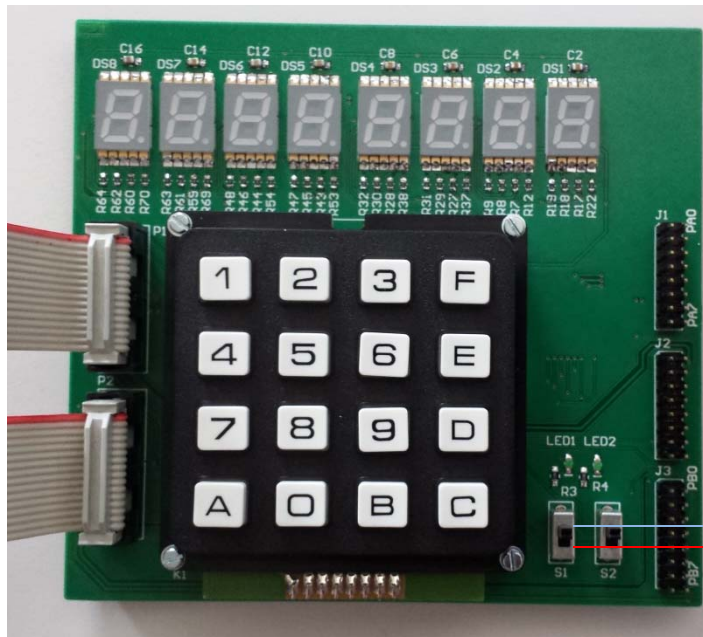
Enable			Input			Output							
LE	E1	E2	A0	A1	A2	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
H	L	H	X	X	X	stable							
X	H	X	X	X	X	L	L	L	L	L	L	L	L
X	X	L	X	X	X	L	L	L	L	L	L	L	L
L	L	H	L	L	L	H	L	L	L	L	L	L	L
			H	L	L	L	H	L	L	L	L	L	
			L	H	L	L	L	H	L	L	L	L	
			H	H	L	L	L	L	H	L	L	L	
			L	L	H	L	L	L	L	H	L	L	
			H	L	H	L	L	L	L	L	H	L	
			L	H	H	L	L	L	L	L	H	L	
			H	H	H	L	L	L	L	L	L	L	

[1] H = HIGH voltage level; L = LOW voltage level; X = don't care; Z = high-impedance OFF-state.

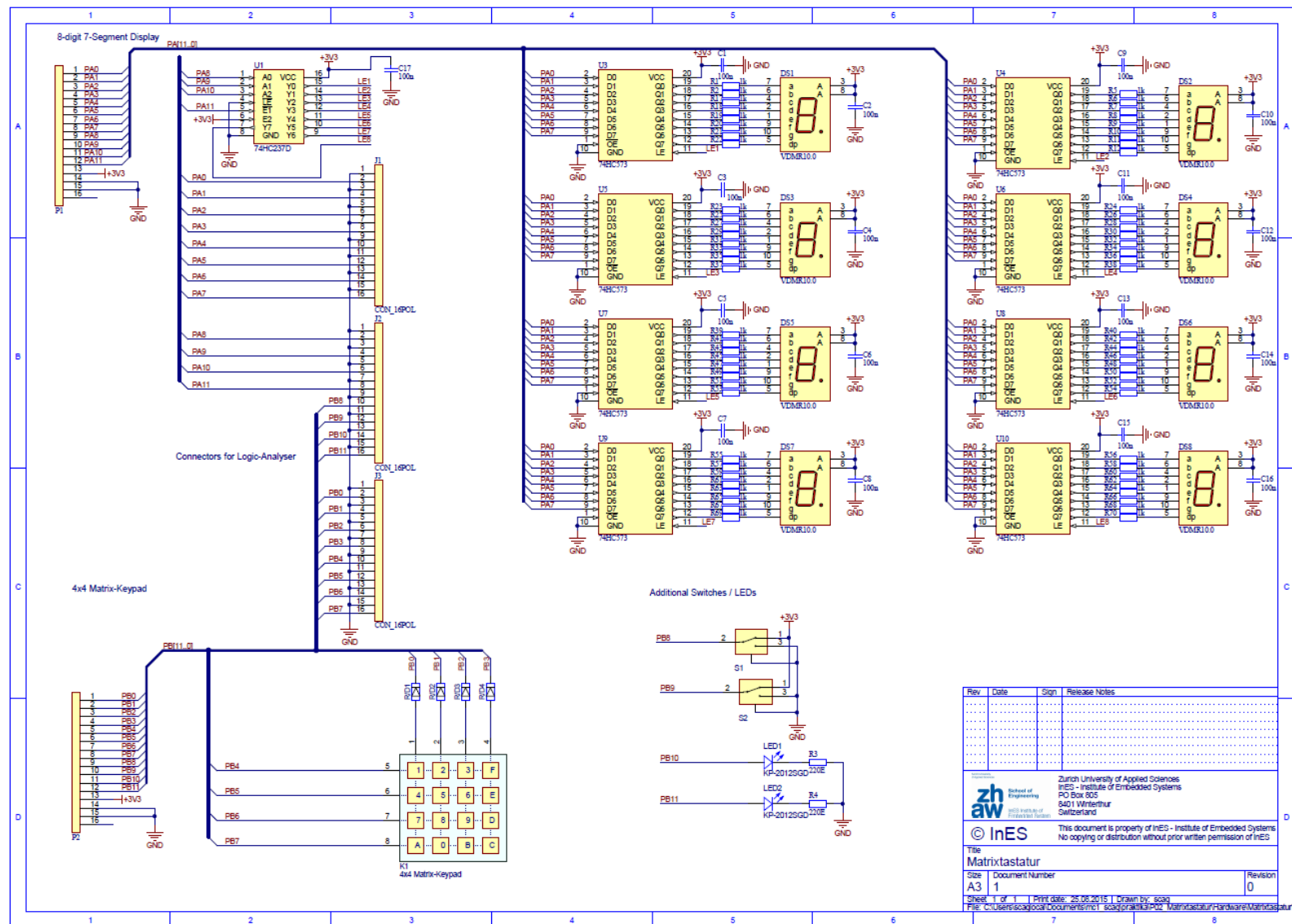
Quelle:

NXP (2015) : 74HC237. URL: [http://www.nxp.com/documents/data\\_sheet/74HC237.pdf](http://www.nxp.com/documents/data_sheet/74HC237.pdf) [Stand: 28.08.2015]

### 4.3 Anschluss ans CT-Board



## 4.4 Schema



### Abbildung 3: Schema Matrixtastatur