

# Eingabe mit Folientastatur

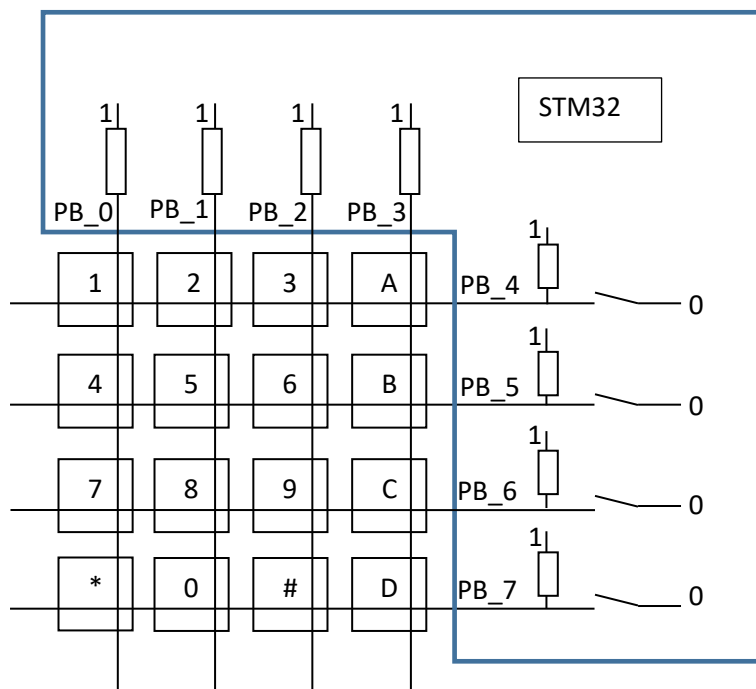


Abbildung 1 Anschlusschema

Beschreibung: Die Folientastatur hat 4 horizontale und 4 vertikale Leitungen. Über den Kreuzungspunkten sitzen die Tasten. Ein Tastendruck bewirkt eine Verbindung beider Leitungen. Die horizontalen und vertikalen Leitungen sind mit den Ports des Mikrocontrollers verbunden.

PB\_0..PB\_3 werden als PortIn spalten mit PullUp konfiguriert.

PB\_4..PB\_7 werden als PortInOut zeilen OpenDrain und output() konfiguriert.

PC\_0..PC\_7 werden, zu Testzwecken, als PortOut pc konfiguriert

Dokumentieren Sie die erforderlichen Programmzeilen.

Beantworten Sie zunächst folgende Fragen:

1. Wenn an zeilen=0b11101111 ausgegeben wird und spalten=0b1110 eingelesen wird, welche Taste wurde dann gedrückt?
2. Wenn an zeilen=0b10111111 ausgegeben wird und spalten=0b1011 eingelesen wird, welche Taste wurde dann gedrückt?
3. Welchen Wert hat spalten wenn zeilen=0b0111 1111 und Taste # gedrückt ist?

Aufgaben:

1. Erstellen Sie das Programm:
  - a. Neues Projekt „Matrixtastatur“ anlegen  
GPIOs manuell konfigurieren entsprechend der Beschreibung oben
2. Tastatur testen.  
Schließen Sie die Tastatur an. In der Endlosschleife:

zeilen= Wert aus der Tabelle unten;  
pc=zeilen+spalten;

Vervollständigen Sie zunächst die Funktionstabelle:

zeilen	Taste	pc=Zeilen+spalten
0b1110 1111	Keine	0b1110 1111
	1	0b1110 1110
	2	0b1110 1101
	3	0b1110 1011
	A	0b1110 0111
0b1101 1111	Keine	0b1101 1111
	4	0b1101 1110
	5	0b1101 1101
	6	
	B	
0b1011 1111	Keine	
	7	
	8	
	9	
	C	
	Keine	
	*	
	0	
	#	
	D	

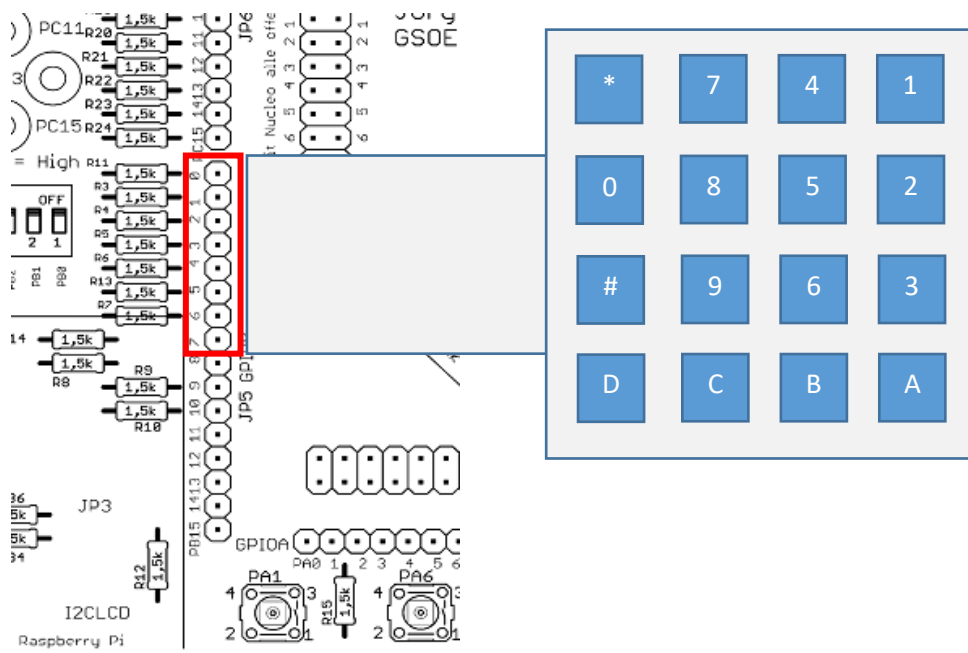


Abbildung 2 Anschluss der Folientastatur

3. Überlegen Sie sich einen Algorithmus zum „Abscannen“ der Tastatur
4. Überlegen Sie, wie die Dekodierung der Werte von zeilen+spalten erfolgen kann
5. Präsentieren und Diskutieren Sie Ihre Vorschläge mit Ihren Mitschülern
6. Vervollständigen Sie Ihr Projekt
  - a. Bibliothek einbinden (siehe Anhang)
  - b. Unterprogramm void ausgabe(unsigned int i) {...}
  - c. Tastatur abscannen in der EndlosschleifeHinweis: Nach der Ausgabe auf zeilen muss 20ms gewartet werden (wait\_ms(20);), bis stabile Werte an spalten verfügbar sind.

Beispiel:

```
#include "LCD.h"

lcd mylcd;

void ausgabe(unsigned int i)
{
}

int main()
{
    // Initialise
    mylcd.clear();
    mylcd.cursorpos(0);
    mylcd.printf("Hallo %c",'x');
    ...
    while (true) {
        zeilen=0b11101111;
        wait_ms(20);
        ausgabe(zeilen+spalten);
        ...
    }
}
```

Anhang:

Hinweise zur Programmierung:

Da die Ausgabe auf das LCD-Display erfolgen soll benötigen Sie folgende Library:

Ergänzen Sie in der main.cpp an den entsprechenden Stellen:

The screenshot shows the Mbed IDE interface. At the top, the 'Import' button is highlighted with a red box and an arrow. Below it, the 'Import Wizard' dialog is open, showing the 'Libraries' tab. The search bar contains 'LCD i2c GSOE', and the search results list the 'LCD i2c GSOE' library by Jack Hansdampf. A red box labeled 'Doppelklick' (double-click) points to the library name in the list. A red box labeled 'Library LCD\_i2c\_GSOE suchen' (search for LCD\_i2c\_GSOE library) points to the search bar. Below the wizard, the 'main.cpp' file is open, showing the following code:

```
1 /* mbed Microcontroller Library
2  * Copyright (c) 2019 ARM Limited
3  * SPDX-License-Identifier: Apache-2.0
4  */
5
6 #include "mbed.h"
7 #include "platform/mbed_t
8 #include "LCD.h"
9
10 int main()
11 {
12     // Initialise
13     mylcd.clear();
14     mylcd.cursorpos(0);
15     mylcd.printf("Hallo %c", 'x');
16     PortIn spalten(PortB, 0b1111);
17     spalten.mode(PullUp);
18 }
```

A red box labeled 'Library mit #include „LCD.h“ einbinden' (include library with #include "LCD.h") points to the line `#include "LCD.h"`. Another red box labeled 'LCD-Display-Objekt mit Namen z.B. mylcd deklarieren und erzeugen' (declare and create LCD display object with name e.g. mylcd) points to the line `lcd mylcd;`. A red box labeled 'Befehle:' (Commands:) points to the code block starting with `int main()`. The commands listed are:

- `mylcd.clear();` //löscht das Display
- `mylcd.cursorpos(wert);` //plaziert den Cursor 0..15: 1.Zeile, 64..79 2.Zeile (64=0x40)
- `mylcd.printf(Formatstring,Werte);` //Ausgabe. Doku: Internet printf