Eingabe mit Folientastatur

1

1

2

1

3

1

A

1

1

4

5

6

B

1

7

8

9

C

1

\*

0

#

D

1

PB\_7

PB\_6

PB\_5

PB\_4

PB\_0

PB\_1

PB\_2

PB\_3

0

0

0

0

STM32

Abbildung 1Anschlussschema



Beschreibung: Die Folientastatur hat 4 horizontale und 4 vertikal 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:

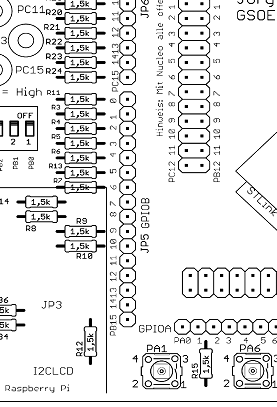
1. Wenn an zeilen=0b11101111 ausgegeben wird und spalten=0b1110 eingelesen wird, welche Taste wurde dann gedrückt?
2. Wenn an zeilen=0b10111111 ausgegeben wird und spalten=0b1011 eingelesen wird, welche Taste wurde dann gedrückt?
3. Welchen Wert hat spalten wenn zeilen=0b0111 1111 und Taste # gedrückt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

Aufgaben:

1. Erstellen Sie das Programm:
   1. Neues Projekt „Matrixtastatur“ anlegen
      1. Ohne Konfiguration der GPIO. Diese soll direkt mit den Registern erfolgen
   2. 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 |  |



A

B

C

D

#

9

6

3

2

5

8

0

1

4

7

\*

Abbildung 2Anschluss der Folientastatur

1. Überlegen Sie sich einen Algorithmus zum „Abscannen“ der Tastatur
2. Überlegen Sie, wie die Dekodierung der Werte von zeilen+spalten erfolgen kann
3. Präsentieren und Diskutieren Sie Ihre Vorschläge mit Ihren Mitschülern
4. Vervollständigen Sie Ihr Projekt
   1. Bibliothek einbinden (siehe Anhang)
   2. Unterprogramm void ausgabe(unsigned int i) {…}
   3. Tastatur abscannen in der Endlosschleife  
      Hinweis: 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:



Doppelklick

Library LCD\_i2c\_GSOE suchen

Import



LCD-Display-Objekt mit Namen z.B. mylcd  
deklarieren und erzeugen

Library mit #include „LCD.h“  
einbinden



Befehle:

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