

Eingabe mit Folientastatur

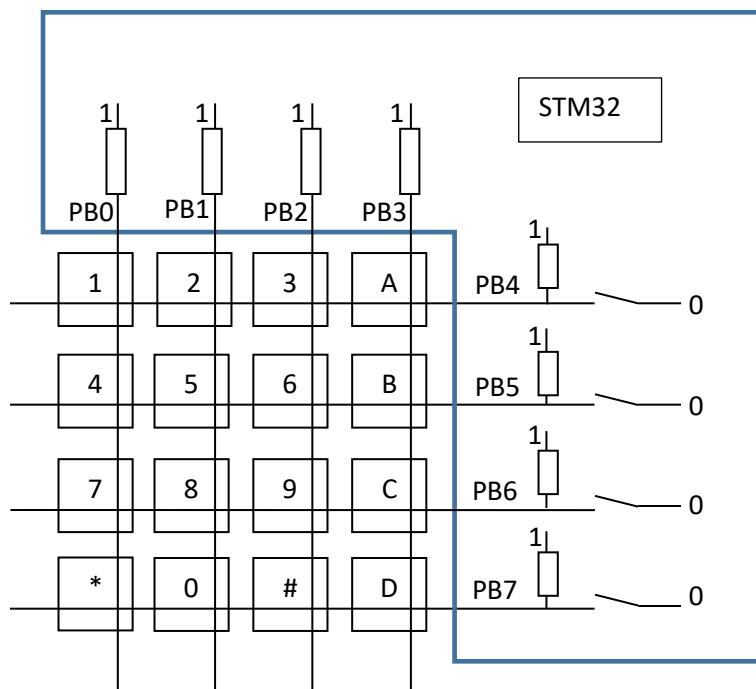


Abbildung 1 Anschlusschema

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.

PB0..PB3 werden als Input mit Pullup konfiguriert.

PB4..PB7 werden als Output Open-Drain mit Pullup konfiguriert.

PC0..PC7 werden als Output Pushpull konfiguriert

Beantworten Sie zunächst folgende Fragen:

1. Wenn an PB4 0 ausgegeben wird und PB0 mit 0 eingelesen wird, welche Taste wurde dann gedrückt?
2. Wenn an PB6 0 ausgegeben wird und PB2 mit 0 eingelesen wird, welche Taste wurde dann gedrückt?
3. Welchen Wert hat GPIOB->IDR wenn GPIOB->ODR=0b0111 1111 und Taste # gedrückt ist?

Aufgaben:

1. Erstellen Sie das Programm:
 - a. Neues Projekt „Matrixtastatur“ anlegen
 - i. Mit Takteinstellung 32MHz
 - ii. Ohne Konfiguration der GPIO. Diese soll direkt mit den Registern erfolgen
 - b. GPIOs manuell konfigurieren entsprechend der Beschreibung oben

2. Tastatur testen.

Schließen Sie die Tastatur an. In der Endlosschleife:

GPIOB->ODR= Wert aus der Tabelle unten;

GPIOC->ODR=GPIOB->IDR;

Vervollständigen Sie zunächst die Funktionstabelle:

GPIOB->ODR	Taste	GPIOB->IDR
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	

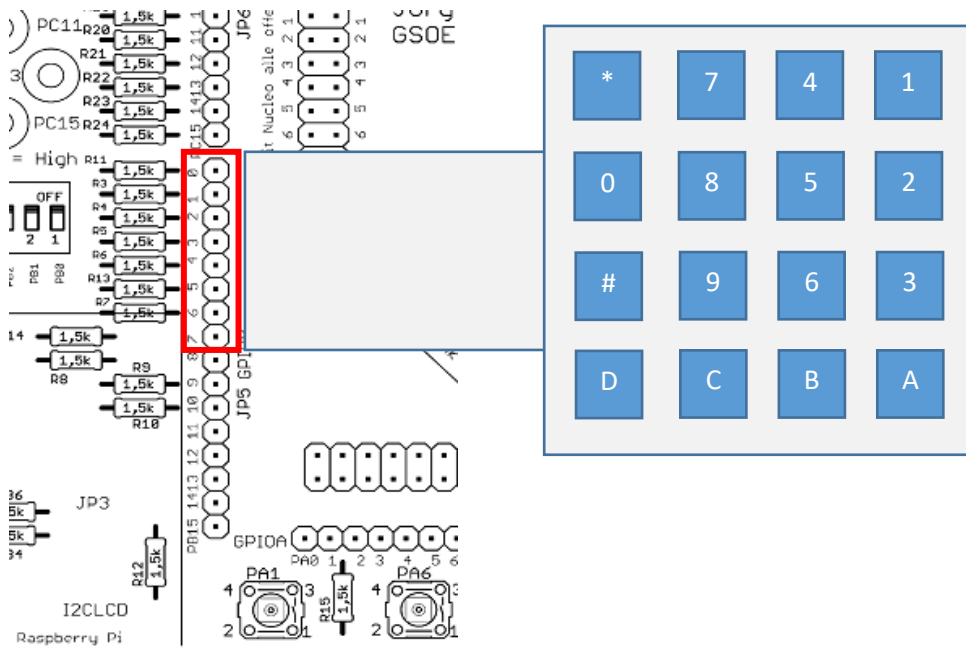


Abbildung 2 Anschluss der Folientastatur

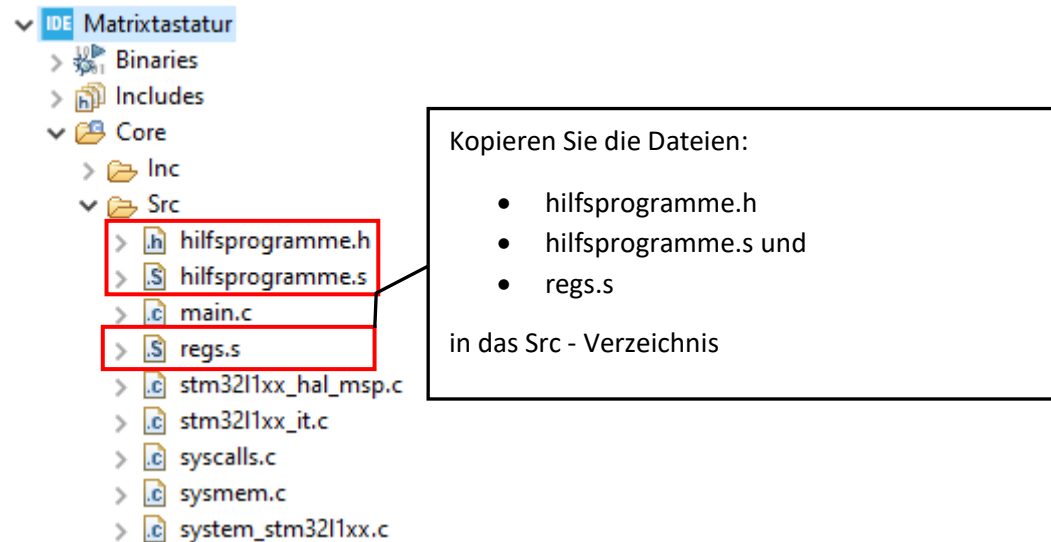
3. Überlegen Sie sich einen Algorithmus zum „Abscannen“ der Tastatur
4. Überlegen Sie, wie die Dekodierung der Werte von GPIOB->IDR erfolgen kann
5. Präsentieren und Diskutieren Sie Ihre Vorschläge mit Ihren Mitschülern
6. Vervollständigen Sie Ihr Projekt
 - a. Hilfsprogramme einbinden
 - b. Unterprogramm Ausgabe
 - c. Tastatur abscannen in der Endlosschleife

Hinweis: Nach der Ausgabe auf GPIOB->ODR muss 20ms gewartet werden, bis stabile Werte an GPIOB->IDR verfügbar sind.

Anhang:

Hinweise zur Programmierung:

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



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

```
...
/* Includes -----
---*/
#include "main.h"

/* Private includes -----
---*/
/* USER CODE BEGIN Includes */
#include "hilsfprogramme.h"
/* USER CODE END Includes */
...
```

Sowie in int main(void):

```
/* Initialize all configured peripherals */
MX_GPIO_Init();
/* USER CODE BEGIN 2 */
//Hier Ihre GPIO Konfiguration

startup();           //LDC-Display initialisieren
LCD_i2c_clear();     //LCD-Display Anzeige löschen
/* USER CODE END 2 */

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{ ...
```

Die Dekodierung und Ausgabe kann mit einem Unterprogramm erfolgen:

```
/* Private user code -----
---*/
/* USER CODE BEGIN 0 */
/*
 * Anschluss der Tastatur an PB0 .. PB7
 */

void Ausgabe(unsigned int i) //i: Wert aus GPIOB->IDR
{
    char z=0xff;

    switch(i)
    {
        case 0xEE: z='1'; break;
        ...
    }
    if (z !=0xff)
    {
        LCD_i2c_cursorpos(0); //Cursor an den Anfang des Displays
        LCD_i2c_asciaus(z); //Zeichen ausgeben
    }
}

/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    /* USER CODE BEGIN 1 */
        unsigned int i=0;
    /* USER CODE END 1 */
    ...
}
```