

Projekt Taschenrechner mit LED-Keyboard und LCD-Display

Technologieschema

Tastenfunktionen:

PB4	Op1plus	Operand 1 um 1 erhöhen
PB3	Op1minus	Operand 1 um 1 erniedrigen
PB0	Op2plus	Operand 2 um 1 erhöhen
PA1	Op2minus	Operand 2 um 1 erniedrigen
PA6	Auswahl	Operation wählen + → - → x → : → + → ...
PA10	Rechne	Berechnen

Funktion: Zahlen von 0 bis ... sollen addiert, subtrahiert, multipliziert und dividiert werden können.

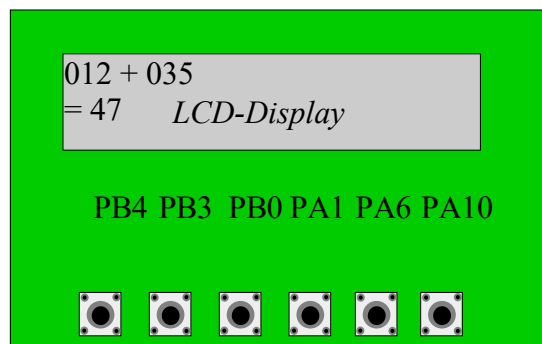
Mit den Tasten PB4 und PB3 soll Operand 1 ausgewählt werden.

Mit den Tasten PB0 und PA1 soll Operand 2 ausgewählt werden.

Mit der Taste PA6 wird zyklisch die Rechenoperation ausgewählt.

Mit Druck auf die Taste PA10 erfolgt die Berechnung.

Hinweise: Die Schalter des Mäuseklaviers müssen alle ausgeschaltet sein (off).



Programmrumpf:

```
.equ Op1plus,Bit4    //GPIOB
.equ Op1minus,Bit3   //GPIOB
.equ Op2plus,Bit0     //GPIOB
.equ Op2minus,Bit1    //GPIOA
.equ Auswahl,Bit6     //GPIOA
.equ rechne,Bit10     //GPIOA

main:
    bl    startLCD
    bl    LCD_i2c_clear
    mov    R1,#0 //1. Zahl
    mov    R2,#0 //2. Zahl
    mov    R3,#0 //Operation: 0='+', 1='-', 2='x', 3=':'
    mov    R4,#0 //Ergebnis
    ldr    R5,=GPIOA
    ldr    R6,=GPIOB

schleife:
//Ausgabe:
    mov    R0,#0
    bl    LCD_i2c_cursorpos
    mov    R0,R1
    bl    LCD_i2c_dezaus
    cmp    R3,#0
    beq    zplus
    cmp    R3,#1
    beq    zminus
    cmp    R3,#2
    beq    zmal
    mov    R0,#'/'
    b      zweiter
zmal:
    mov    R0,#'*'
    b      zweiter
zminus:
    mov    R0,#'-'
    b      zweiter
zplus:
    mov    R0,#'+'
zweiter:
    bl    LCD_i2c_asciaus
    mov    R0,R2
    bl    LCD_i2c_dezaus
    ldr    R0,=leer
    bl    LCD_i2c_textaus
    mov    R0,#0x40
    bl    LCD_i2c_cursorpos
    mov    R0,R4
    bl    LCD_i2c_dezaus
    ldr    R0,=leer
    bl    LCD_i2c_textaus
//Ende Ausgabe
//Hier Ihr Code
    b      schleife

leer:
.asciz " "
.end
```

Aufgaben:

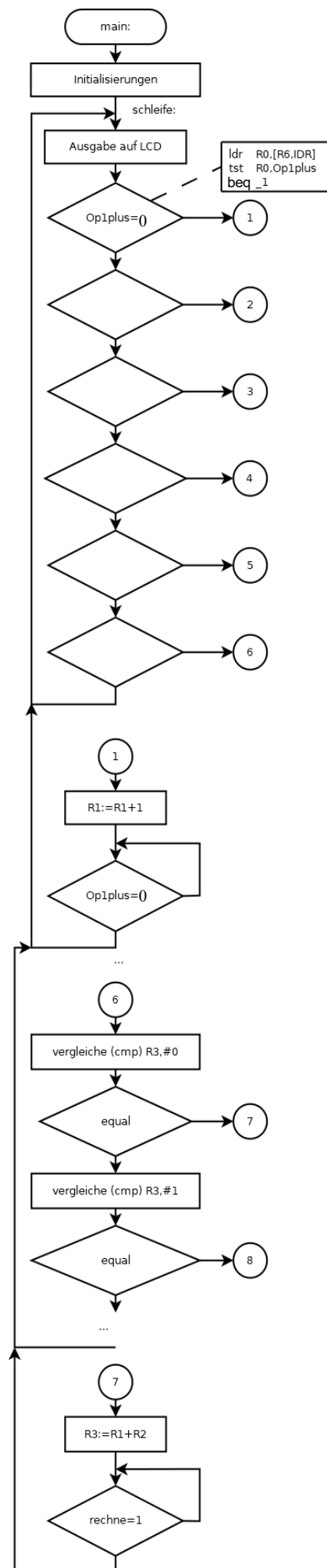
1. Programmablaufplan PAP erstellen
In der Endlosschleife müssen folgende Aufgaben erfüllt werden:
 1. Taste **Op1plus** abfragen, R1 gegebenenfalls erhöhen
 2. Taste **Op1minus** abfragen, R1 gegebenenfalls verringern
 3. Ebenso mit den Tastern **Op2plus** und **Op2minus** mit R2
 4. Taste **Auswahl** abfragen, R3 zyklisch von 0 bis 3 durchzählen.
 5. Taste **rechne** abfragen, gegebenenfalls R4 neu berechnen.
 6. Ausgabe aller Zahlen auf das Display.(Unterprogramm **Anzeige:**)
2. Programm schreiben und testen.

Hinweise zur Verwendung des LCD-Displays:

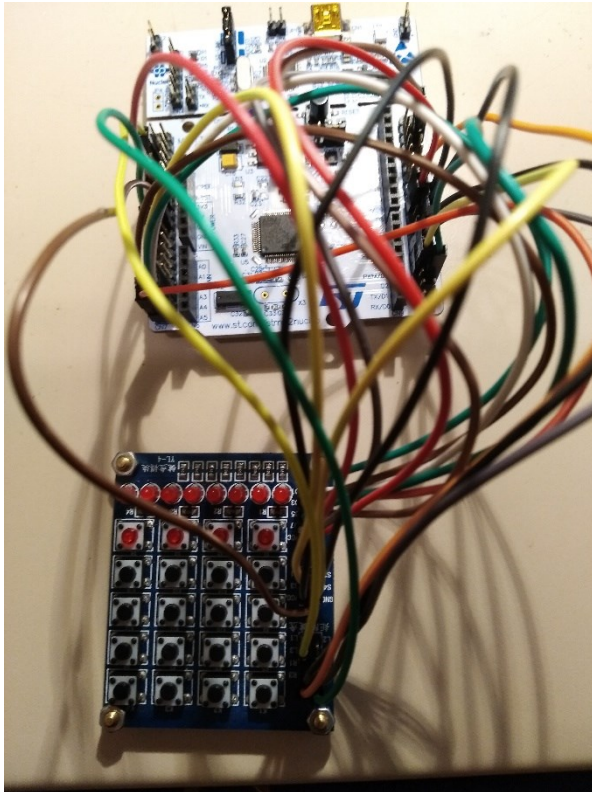
1. `bl LCD_i2c_clear` //löscht das Display.
2. Cursor positionieren:
`mov R0,#0xaa` // aa = Cursorposition: 1. Zeile 0 .. 3Fh, 2. Zeile 40h .. 7Fh
`bl LCD_i2c_cursorpos`
3. R0 als Dezimalzahl auf die aktuelle Cursorposition ausgeben
`bl LCD_i2c_dezaus`
3. R0 als ASCII-Zeichen auf die aktuelle Cursorposition ausgeben:
`mov R0,#'+'`
`bl LCD_i2c_asciaus`

weitere Operationen: Siehe Formelsammlung

Hilfe PAP:



Ein- und Ausgabe mit LED-Keyboard und LCD-Display



Nucleo	LCD
CN7 Pin 18 +5V	VCC
CN7 Pin 20 GND	GND
CN10 Pin 12 PA12	SDA GPIO-Output Opendrain Pullup Highspeed
CN10 Pin 14 PA11	SCL GPIO-Output Highspeed
	LED Keyboard
CN7 Pin 16 3,3V	VCC
CN7 Pin 34 PB0	S1 GPIO-Input kein Pull
CN10 Pin 24 PB1	S2 GPIO-Input kein Pull
CN10 Pin 22 PB2	S3 GPIO-Input kein Pull
CN10 Pin 31 PB3	S4 GPIO-Input kein Pull
CN7 Pin 20 GND	GND
CN7 Pin 19 GND	L1
CN10 Pin 33 PA10	R1 GPIO-Input Pullup
CN10 Pin 13 PA6	R2 GPIO-Input Pullup
CN7 Pin 30 PA1	R3 GPIO-Input Pullup
CN10 Pin 27 PB4	R4 GPIO-Input Pullup

Anschlussbelegung und Konfiguration:
 Beachten Sie: Die LEDs sind Lowaktiv!!
 Eventuell **mvn R0,R0** Instruktion zur Negation verwenden.

