Codeschloss mit Zustandsdiagramm

Technologieschema:

7

PC\_0..PC\_6

„1“

PA\_6

takt

Mikrocontroller

PA\_1

„1“

s

freigabe

PC\_7

a

g

f

b

e

c

d

Lösung mit Mikrocontroller

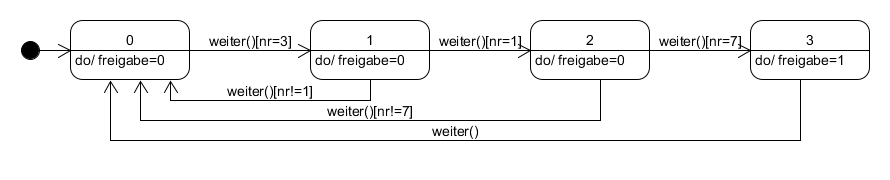
Siebensegmenttabelle:  
int seg7[10]={0b00111111, 0b00000110, 0b01011011, 0b01001111, 0b01100110,

0b01101101, 0b01111101, 0b00000111, 0b01111111, 0b01101111};

* 3-stelliger Code (in dieser Aufgabe: 3-1-7)
* Das Einstellen der der Ziffern erfolgt mit Hilfe eines prellfreien Tasters „s“, mit dessen Hilfe auf der Siebensegmentanzeige die Ziffern 0..9 zyklisch „durchgetastet“ werden können.
* Die Übernahme einer Ziffer erfolgt mit dem Taster „Takt“
* Für den Fall, dass die Kombination richtig war (hier 3-1-7) wechselt der Pegel von PC\_7 freigabe von 0 nach 1.
* Taster mit PullDown

Aufgaben:

1. Zifferneingabe mit Taster „s“. Bei jedem Interrupt wird die globale Variable int nr zyklisch von 0 bis 9 „durchgetastet“ und auf der Siebensegmentanzeige dargestellt.  
   Schreiben Sie die Initialisierung und die ISR in C.
2. Die Funktion der Taste „takt“ wurde in einem Zustandsdiagramm entwickelt:  
   Die globale Zustandsvariable int zustand; kann die Werte 0 .. 3 annehmen.  
   Programmieren Sie die Initialisierung und die ISR des Interrupts von takt (void weiter()).



1. Wenn 5 Sekunden lang keine Taste gedrückt wurde, dann soll die Anzeige verlöschen und der Mikrocontroller in Zustand 0 wechseln. Gefordert: Erweiterung des Zustandsdiagramms, Initialisierung und ISR des Timers mit Interrupt

Lösung:

#include "mbed.h"

#include "platform/mbed\_thread.h"

#include "LCD.h"

InterruptIn s(PA\_1);

InterruptIn takt(PA\_6);

PortOut anzeige(PortC,0x7F);

DigitalOut freigabe(PC\_7);

DigitalOut seg7On(PC\_15);

lcd mylcd;

int zustand=0;

int nr=0;

int seg7[10]={0b00111111, 0b00000110, 0b01011011, 0b01001111, 0b01100110,

0b01101101, 0b01111101, 0b00000111, 0b01111111, 0b01101111};

void hoch()

{

nr=(nr+1)%10;

anzeige=seg7[nr];

TIM6->CNT=0;

}

void weiter()

{

switch(zustand)

{

case 0: if (nr==3) zustand=1; break;

case 1: if (nr==1) zustand=2; else zustand=0; break;

case 2: if (nr==7) zustand=3; else zustand=0; break;

default: zustand=0;

}

}

void ruecksetz()

{

zustand=0;

anzeige=0;

TIM6->SR=0;

HAL\_NVIC\_ClearPendingIRQ(TIM6\_IRQn);

}

void init(void)

{

s.mode(PullDown);

takt.mode(PullDown);

s.rise(&hoch);

takt.rise(&weiter);

mylcd.clear();

freigabe=0;

seg7On=1;

RCC->APB1ENR|=0b10000; //Clock Enable

TIM6->PSC=31999; //Prescaler 1ms

TIM6->ARR=4999; //Autoreload 5000\*1mss = 5s

TIM6->DIER=1; //UIE = 1 (Update Interrupt Enable)

TIM6->SR=0; //UIF =0 (Update Interrupt Flag)

TIM6->CR1=1; //CEN=1 (Counter Enable)

/\* TIM6\_IRQn interrupt configuration \*/

NVIC\_SetVector(TIM6\_IRQn, (uint32\_t)&ruecksetz);

HAL\_NVIC\_EnableIRQ(TIM6\_IRQn);

}

int main()

{

init();

while (true) {

if (zustand==3) freigabe=1;

else freigabe=0;

mylcd.cursorpos(0);

mylcd.printf("Zustand= %d",zustand);

}

}