

HÖHERE TECHNISCHE BUNDESLEHRANSTALT HOLLABRUNN

Höhere Abteilung für Elektronik – Technische Informatik

Klasse / Jahrgang: 4BHEL	Gruppe: 4b	Übungsleiter: Dipl.-Ing. Josef Reisinger
Übungsnummer: III / 5	Übungstitel: Mikroprozessor Interfacetechnik	
Datum der Übung: 10.02.2015	Teilnehmer: Graussam, Holzinger, Mischling, Schwarzmann	
Datum der Abgabe: 24.02.2015	Schrittführer: Graussam	Unterschrift:

	Beurteilung
Aufgabenstellung	
Dokumentation	
Messschaltungen	
Messtabellen	
Berechnungen	
Programmlistings	
Auswertung	
Diagramme	
Berechnungen	
Simulationen	
Schlussfolgerungen	
Kommentare	
Inventarliste	
Messprotokoll	
Form	
Summe	

1 Allgemeiner Teil

Titel der Übung: Mikroprozessor Interfacetechnik
Übungsnummer: III / 5
Übungsplatz: T10
Datum der Übung: 10.02.2015
Klasse: 4BHEL
Schriftführer: Lukas Graussam
Übungsteilnehmer: Lukas Graussam
Thomas Holzinger
David Mischling
Philipp Schwarzmann

Inhaltsverzeichnis:

1 Allgemeiner Teil	2
2 Besprechung	3
3 Aufgabenstellung	3
4 Messschaltung	3
4.1 Informationen zur Schaltung und Funktion	4
5 Software	4
6 Messungen und Auswertungen	8
7 Erkenntnisse	9
Inventarliste	10
Messprotokoll	11

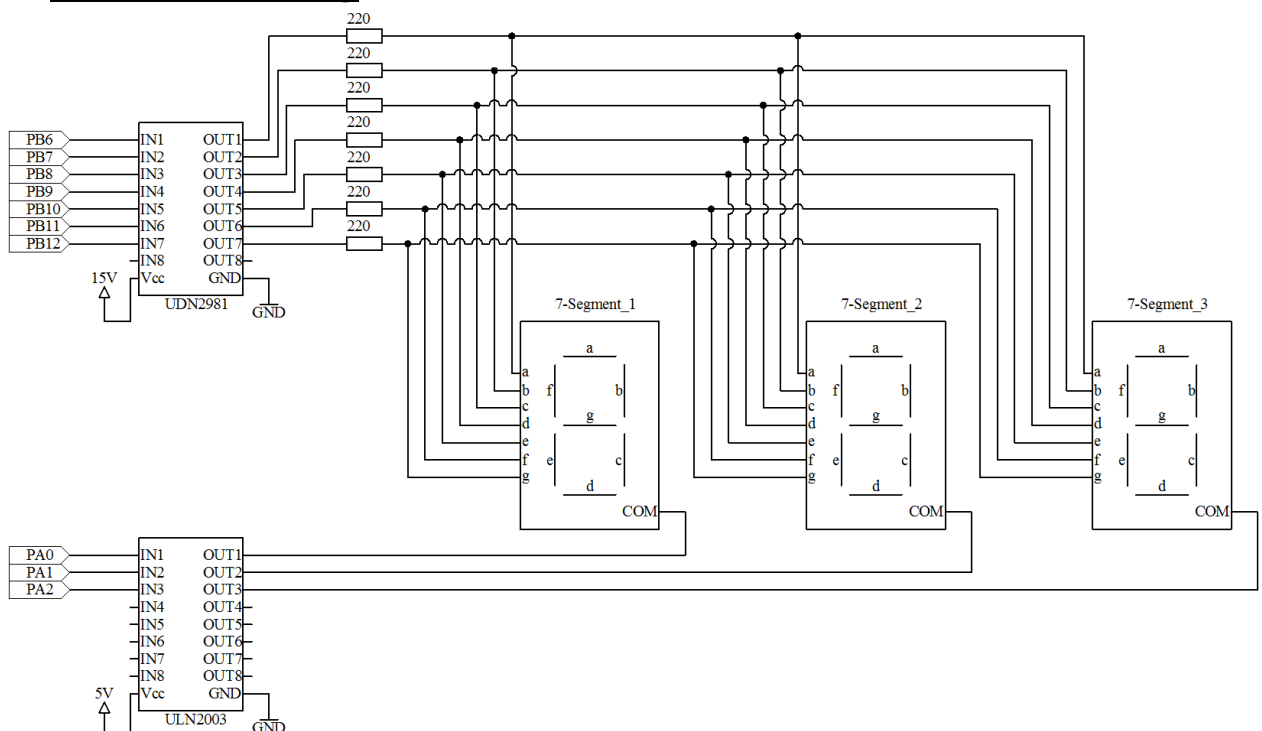
2 Besprechung

Vor der Übung wurden alle wichtigen Informationen zur Interfacetechnik bei Mikroprozessoren besprochen. Vor allem der Multiplexbetrieb bei 7-Segment Anzeigen (auch Scan genannt) mit den Bausteine UDN und ULN wurde detailliert wiederholt. Diese beiden Bausteine bestehen vereinfacht erklärt aus Darlington-Transistoren, welche auf V_{CC} (UDN) bzw. Masse (ULN) schalten, sobald eine logische „1“ am jeweiligen Eingang angelegt wird. Genauer ist im jeweiligen Datenblatt nachzulesen.

3 Aufgabenstellung

Die Aufgabe war es nun, einen Scan bei einer dreistelligen 7-Segment Anzeige zu realisieren. Die Bausteine ULN 2003 und UDN 2981, sowie die 7-Segment Anzeigen wurden uns dafür zur Verfügung gestellt. Auf einem Steckbrett sollte die Hardware aufgebaut werden, und die Ansteuerung sollte mit einem Programm für den Cortex M3 realisiert werden. Danach sollten auf jede Stelle der 7-Segment Anzeige verschiedene Zeichen ausgegeben werden.

4 Messschaltung



4.1 Informationen zur Schaltung und Funktion

Informationen zum Scan

Die 7-Segment Anzeigen, welche wir verwendeten, hatten eine gemeinsame Kathode (COM). Deshalb wurde der ULN für den Multiplexbetrieb verwendet (um nacheinander die drei Anzeigen zu aktivieren). Dies muss so schnell geschehen, dass das menschliche Auge ein fixes Bild sieht. Wenn man annimmt, dass das Auge ca. 30Hz „braucht“, benötigt man bei drei Anzeigen das dreifache, also 90Hz. Wir entschieden uns bei der Durchführung mit 100Hz zu arbeiten.

Verwendete Portleitungen

Für die Verbindung des SUB D 25 Stecker des Cortex M3 zum Steckbrett wurde uns ein Bus-Kabel zur Verfügung gestellt, bei dem nicht alle Leitungen verbunden waren. PA0-2 (für Scan) und PB6-12 (für Segmente) wurden von uns verwendet, da diese verbunden waren.

Widerstände

Für die Anzeigen mussten Vorwiderstände verwendet werden. Aus Erfahrung nahmen wir dafür 220Ω.

5 Software

Wir entschieden uns dazu, das Programm für den Cortex in Assembler zu schreiben. Der Code befindet sich auf der nächsten Seite (Anmerkungen folgen nach dem Code).

```

/*****
/*
/* File Name:   Scan.c
/* Autor:      Lukas Graussam
/* Version:    V1.00
/* Date:       10/02/2015
/* Description: Multiplexbetrieb - 7-Segment-Anzeigen
*****/
/* History:    V1.00
*****/

```

```

AREA Scan, CODE, READONLY
INCLUDE STM32_F103RB_MEM_MAP.INC
EXPORT __main

```

;Bitbanding für jede benötigte Portleitung:

```

Seg_A      EQU    PERIPH_BB_BASE+(GPIOB_ODR - PERIPH_BASE)*0x20+6*4
Seg_B      EQU    PERIPH_BB_BASE+(GPIOB_ODR - PERIPH_BASE)*0x20+7*4
Seg_C      EQU    PERIPH_BB_BASE+(GPIOB_ODR - PERIPH_BASE)*0x20+8*4
Seg_D      EQU    PERIPH_BB_BASE+(GPIOB_ODR - PERIPH_BASE)*0x20+9*4
Seg_E      EQU    PERIPH_BB_BASE+(GPIOB_ODR - PERIPH_BASE)*0x20+10*4
Seg_F      EQU    PERIPH_BB_BASE+(GPIOB_ODR - PERIPH_BASE)*0x20+11*4
Seg_G      EQU    PERIPH_BB_BASE+(GPIOB_ODR - PERIPH_BASE)*0x20+12*4

Scan0      EQU    PERIPH_BB_BASE+(GPIOA_ODR - PERIPH_BASE)*0x20+0*4
Scan1      EQU    PERIPH_BB_BASE+(GPIOA_ODR - PERIPH_BASE)*0x20+1*4
Scan2      EQU    PERIPH_BB_BASE+(GPIOA_ODR - PERIPH_BASE)*0x20+2*4

```

```

;*****
/*
/*      HAUPTPROGRAMM:   main
/*
/*
/* Aufgabe:   Aktiviert Unterschiedliche Segmente der jeweiligen Anzeigen
;*****

```

```

__main      BL    init_port
__main_again
    LDR R1, =Seg_C
    MOV R0, #0x1
    STR R0, [R1]
    LDR R1, =Seg_B
    MOV R0, #0x1
    STR R0, [R1]
    LDR R1, =Seg_D      ;Setzt die Segmente B,C,D,F auf 1
    MOV R0, #0x1
    STR R0, [R1]
    LDR R1, =Seg_F
    MOV R0, #0x1
    STR R0, [R1]
    BL _Scan0           ;Erste Anzeige aktivieren
    LDR R1, =Seg_C
    MOV R0, #0x1
    STR R0, [R1]
    LDR R1, =Seg_B
    MOV R0, #0x1
    STR R0, [R1]
    LDR R1, =Seg_D      ;Setzt die Segmente B,C auf 1
    MOV R0, #0x0
    STR R0, [R1]
    LDR R1, =Seg_F
    MOV R0, #0x0
    STR R0, [R1]
    BL _Scan1           ;Zweite Anzeige aktivieren
    ;weiter auf der nächsten Seite

```

```

        LDR R1, =Seg_C
        MOV R0, #0x1
        STR R0, [R1]
        LDR R1, =Seg_B
        MOV R0, #0x0
        STR R0, [R1]
        LDR R1, =Seg_D      ;Setzt die Segmente C,D auf 1
        MOV R0, #0x1
        STR R0, [R1]
        LDR R1, =Seg_F
        MOV R0, #0x0
        STR R0, [R1]
        BL _Scan2           ;Dritte Anzeige aktivieren
        B _main_again      ;Endlosschleife

;*****
;*      U N T E R P R O G R A M M:   Scan 0
;*
;* Aufgabe:   Aktiviert die erste 7-Segment Anzeige
;*
;*****
_Scan0    PUSH {R0-R1, LR}
          MOV R0, #0x01
          LDR R1, =Scan0
          STR R0, [R1]
          BL wait_10ms
          MOV R0, #0x0      ;nach 10ms wird wieder 0 auf der Leitung ausgegeben
          STR R0, [R1]
          POP {R0-R1, PC}

;*****
;*      U N T E R P R O G R A M M:   Scan 1
;*
;* Aufgabe:   Aktiviert die zweite 7-Segment Anzeige
;*
;*****
_Scan1    PUSH {R0-R1, LR}
          MOV R0, #0x01
          LDR R1, =Scan1
          STR R0, [R1]
          BL wait_10ms
          MOV R0, #0x0      ;nach 10ms wird wieder 0 auf der Leitung ausgegeben
          STR R0, [R1]
          POP {R0-R1, PC}

;*****
;*      U N T E R P R O G R A M M:   Scan 2
;*
;* Aufgabe:   Aktiviert die dritte 7-Segment Anzeige
;*
;*****
_Scan2    PUSH {R0-R1, LR}
          MOV R0, #0x01
          LDR R1, =Scan2
          STR R0, [R1]
          BL wait_10ms
          MOV R0, #0x0      ;nach 10ms wird wieder 0 auf der Leitung ausgegeben
          STR R0, [R1]
          POP {R0-R1, PC}

```

```

;*****
;*          U N T E R P R O G R A M M:    wait_10ms          *
;*          *                                                                 *
;* Aufgabe:  Warteschleife für 10ms um 100Hz zu bekommen      *
;*          *                                                                 *
;*****

wait_10ms      push {R0-R2,LR} ; save link register to Stack
               MOV R0,#0x10 ; wait 10ms
               MOV R1,#0
wait_ms_loop   MOV R2,#0x63B
wait_ms_loop1  SUB R2,R2,#1
               CMP R2,R1
               BNE wait_ms_loop1
               SUB R0,R0,#1
               CMP R0,R1
               BNE wait_ms_loop
               POP {R0-R2,PC}

;*****
;*          U N T E R P R O G R A M M:    init_port          *
;*          *                                                                 *
;* Aufgabe:  Initialisiert die benötigten Portleitungen      *
;*          *                                                                 *
;*****

init_port      PUSH {R1-R2, R11, LR}
               MOV R2, #0xC
               LDR R1, =RCC_APB2ENR
               LDR R11, [R1] ;Takt
               ORR R11, R11, R2
               STR R11, [R1]

               LDR R1, =GPIOB_CRH
               LDR R11, [R1] ;B 8, 9, 10, 11, 12
               LDR R2, =0xFFFF0000
               AND R11, R11, R2
               LDR R2, =0x00033333
               ORR R11, R11, R2
               STR R11, [R1]

               LDR R1, =GPIOA_CRL
               LDR R11, [R1] ;A 0, 1, 2
               LDR R2, =0xFFFFF000
               AND R11, R11, R2
               LDR R2, =0x00000333
               ORR R11, R11, R2
               STR R11, [R1]

               LDR R1, =GPIOB_CRL
               LDR R11, [R1] ;B 6, 7
               LDR R2, =0x0FFFFFFF
               AND R11, R11, R2
               LDR R2, =0x33000000
               ORR R11, R11, R2
               STR R11, [R1]

               POP {R1-R2, R11, PC}

END

```

Wegen Zeitmangels kamen wir nicht mehr dazu, sinnvolle Zeichen auszugeben. Anstatt ein Unterprogramm für die Zeichenausgabe zu schreiben, werden in unserem Fall aufgrund des Zeitdrucks im Hauptprogramm nur verschiedene Segmente an den jeweiligen Anzeigen aktiviert. Dies reichte jedoch, um die Funktion des Scans auf den Anzeigen zu erkennen und das Scan-Signal auf den drei Portleitungen PA 0-2 mit dem Oszilloskop zu messen.

6 Messungen und Auswertungen

Oszilloskop (P1) Messung

Hier der Screenshot der Messung der drei Portleitungen PA0-2:



Kommentar: Man kann deutlich die Funktion des Scans erkennen. Es ist immer nur eine Leitung aktiv, gefolgt von der jeweiligen nächsten. Was außerdem auffällt, ist dass die Spannung des zweiten und dritten Signals um einen Faktor 10 kleiner ist als das erste. Dies liegt vermutlich an einem Tastkopffehler.

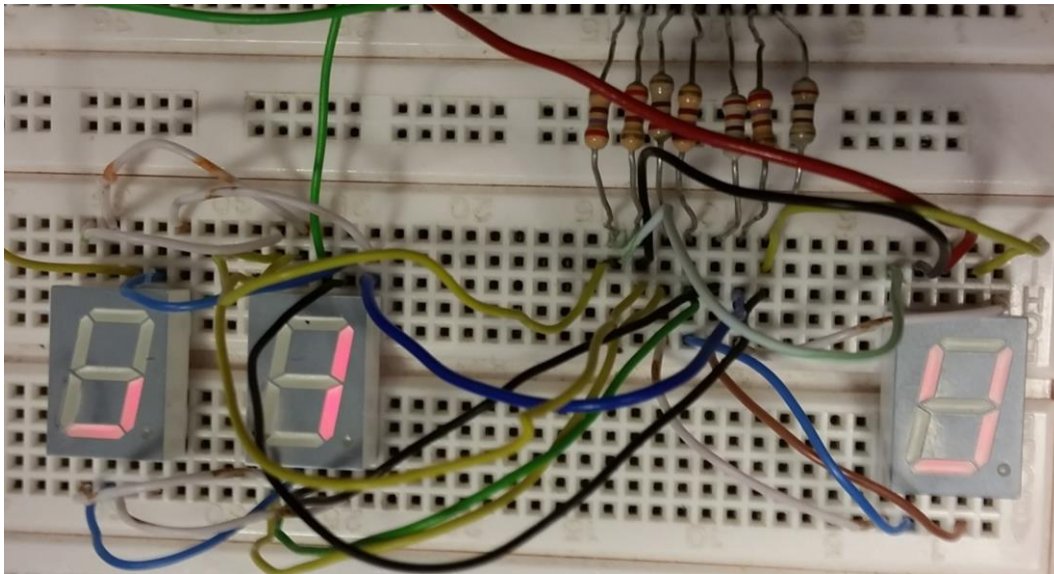
Funktionsbestätigung durch Anzeige

Wie im Programm zu sehen ist, sollten folgende Segmente bei den Anzeigen aktiviert sein:

Anzeige 1: Segment B, C, D und F

Anzeige 2: Segment B und C

Anzeige 3: Segment C und D



Kommentar: Wie das Bild bestätigt, erkennt man bei jeder Anzeige ein anderes „Zeichen“. Anzumerken ist, dass sich auf der rechten Seite die erste Anzeige und auf der linken Seite die dritte Anzeige befindet (vergleiche mit Programm). Außerdem sind bei jeder Anzeige die richtigen Segmente wie im Code beabsichtigt aktiviert, was die Funktion bestätigt.

Wie bereits erwähnt, konnte das Programm aus Zeitmangel nicht weiter optimiert werden, um richtige Zeichen auszugeben.

7 Erkenntnisse

- Multiplexbetrieb: Theoretische Kenntnisse vertieft und praktisch angewendet
- Fähigkeiten zur Assemblerprogrammierung verbessert und aufgefrischt
- Erkenntnisse über eine gute Arbeitsaufteilung in Teams
- Umgang mit Datenblätter und Manuals weiter verbessert

Anmerkung: Das Multimeter P2, welches sich in der Inventarliste befindet, wurde im Endeffekt nicht wirklich verwendet.