

Microcontroller VU 182.694

Exam 1 (Programming)

Familiennamen/Surname:	Vorname/First name:
MatrNr/MatrNo:	Unterschrift/Signature:

Vom Betreuer auszufüllen/To be filled in by supervisor

Funktioniert? Kommentar

Unterschrift d. Betreuers

Bsp 1: Ja Nein

Bsp 2: Ja Nein

Prüfungsablauf

(Please refer to next page for English translation!)

Sie erhalten mit dieser Angabe zwei Programmieraufgaben, für die Sie insgesamt ca. 75 Minuten Zeit haben (ca. 25 Minuten Vorbereitung und ca. 50 Minuten am Rechner).

Sobald ein Beispiel funktioniert, rufen Sie bitte einen Betreuer und führen es vor. Alle Beispiele, die nicht innerhalb der vorgegebenen Zeit vorgeführt wurden, gelten als inkorrekt. Berücksichtigen Sie bitte, dass bei Abgaben in den letzten Minuten möglicherweise gerade kein Betreuer frei ist.

- Bitte füllen Sie zuerst den Testkopf mit Namen und Matrikelnummer aus, und legen Sie Ihren Studentenausweis bereit.
- Falls Sie sich Notizen machen wollen, schreiben Sie bitte nur in die hier vorliegende Angabe an den dafür vorgesehenen Stellen oder auf die Blattrückseiten. Zusatzblätter dürfen nicht verwendet werden.
- Als Unterlagen dürfen nur die von uns gestellten PDF-Dateien verwendet werden.
- Es gibt zu jedem Beispiel eine funktionierende Musterlösung als `ihex` Datei. Damit können Sie überprüfen, ob ein Problem mit der Hardware oder Ihrer Verdrahtung vorliegt. Damit Ihre Lösung von der Musterlösung unterscheidbar ist, blinkt die Musterlösung nach einem Reset auf einigen der verwendeten LEDs. Verwenden Sie zum Assemblieren `avr-as` und `avr-ld` und zum Downloaden Ihres Programms/der Musterlösung `avrprog2`.

Examination Hints

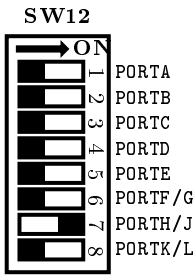
With these instructions, you receive two programming tasks. You have about 75 minutes to program the solutions (approx. 25 minutes for preparation and approx. 50 minutes on the computer).

If one of your solutions is working, please present it to the supervisor immediately. All solutions which are not presented during your working time are noted as incorrect. Please be aware that nobody may be available for assessment during the last minutes of working time.

- Please fill in your name and matriculation number in the header and prepare your student card.
- During the examination, please write any notes only in the provided fields or on the back of these paper sheets. You are not allowed to use additional sheets of paper.
- The provided PDF files are the only documents you are allowed to use.
- For each task a working sample solution is available as `ihex` file. You can use the sample solution to check if there is a problem with the hardware or your wiring. To distinguish from your solution, the sample will flash some of the used LEDs after reset. Use `avr-as` and `avr-ld` to assemble your program. To download your program or the sample solution use `avrprog2`.

A Digital I/O – task1.s

Please set up the switches in the following way:



All other switches are set to off.
Connect *J12* to *GND*.

Create a program, which displays the current state of the buttons 1, 2 and 3 on *PORTB* as given in the following table:

PORTB[1]	PORTB[2]	PORTB[3]	LED_H[1]	LED_H[0]	LED_H[3]
on	*	*	-	⊗	-
off	on	*	⊗	-	-
off	off	on	⊗	⊗	-
off	off	off	-	-	⊗

Remarks: * means don't care (the state of the button is not important). "⊗" means that LEDx is turned on, "-" means that LEDx is turned off.

A.1 Task 1 source

```

; TASK1 - Digital I/O

.NOLIST
.INCLUDE "m1280def.inc"
.LIST

.equ temp, 0x10

; Name the registers you are using.
;*****

```



```
*****

; Program code belongs into text-segment = FLASH
.section    .text

.global    main

.org 0x0000
rjmp     main

; Main
main:
    ; initialize stack pointer
    ldi temp, lo8(RAMEND)
    out SPL, temp
    ldi temp, hi8(RAMEND)
    out SPH, temp

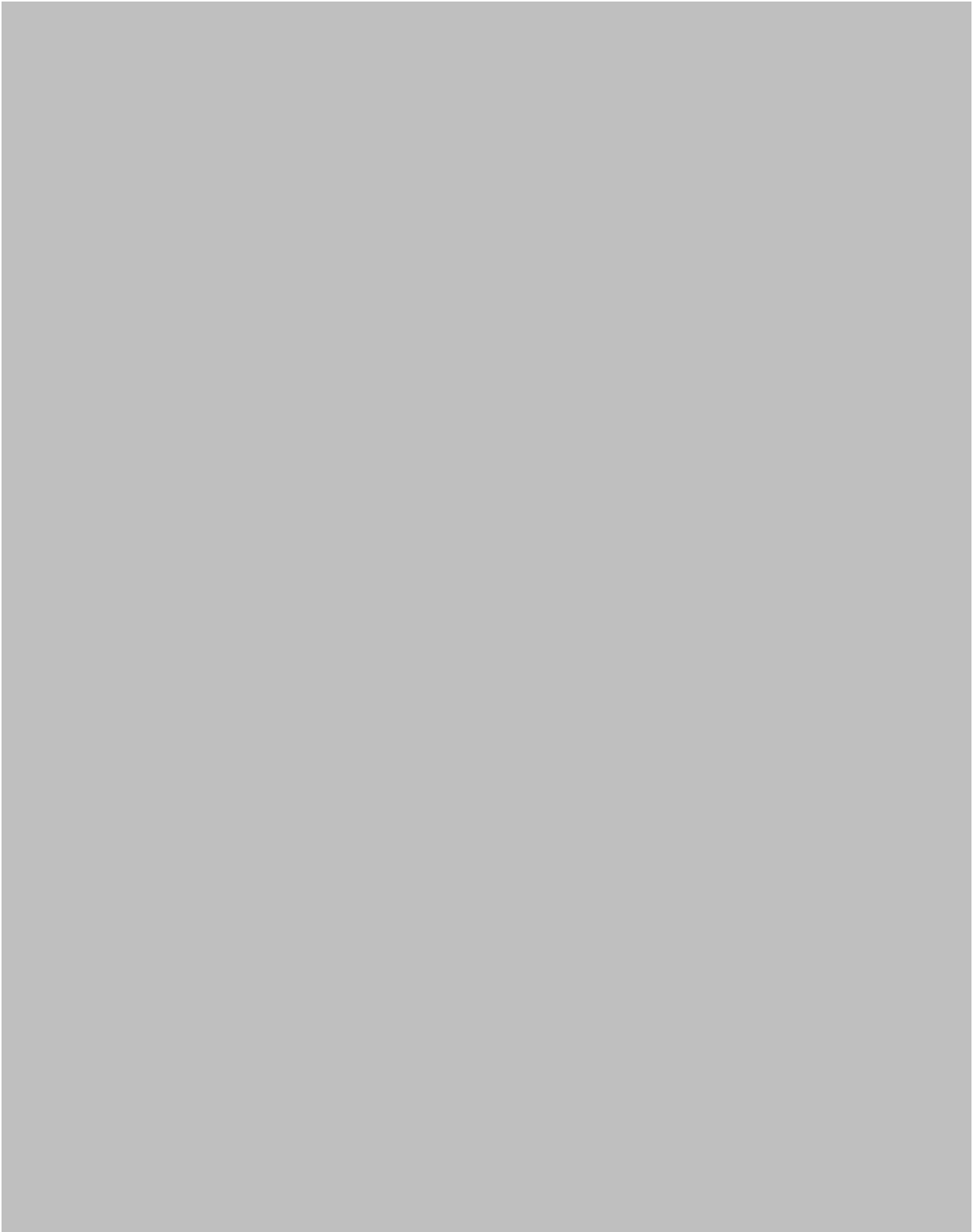
    ; do your initializations here
*****
```


```
*****

infinite_loop:

    ; implement main task here
```

;*****



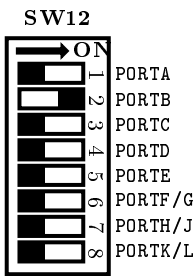


```
*****
```

```
; do the loop  
rjmp infinite_loop
```

B Interrupt - Calling Conventions – task2.s

Please set up the switches in the following way:



All other switches are set to off.

Connect *J12* to *GND*.

Create a program which rotates a LED on *PORTB* whenever the interrupt service routine for INT7 is called. The program should produce the following sequence of patterns:

State-No.	LED_B[0]	LED_B[1]	LED_B[2]	LED_B[3]
1	⊗	-	-	-
2	-	⊗	-	-
3	-	-	⊗	-
4	-	-	-	⊗
5	-	-	⊗	-
6	-	⊗	-	-
7	⊗	-	-	-

The sequence of states should be controlled by the button which corresponds to INT7 in the following way:

1 → **button press** → 2 → button press → 3 → button press → 4 → **button release** → 5 → button release → 6 → button release → 7 → button release → go back to 1 and do it again.

B.1 Task 2 source

```
; TASK2 - Interrupt programming

.NOLIST
.INCLUDE "m1280def.inc"
.LIST

.equ temp, 0x10

; Name the registers you are using.
```

```
;*****
```

```
;*****
```

```
; Program code belongs into text-segment = FLASH  
.section .text
```

```
.global main
```

```
.org 0x0000
```

```
rjmp main
```


```
; install your ISR
```

```
;*****
```

```
;*****
```

```
; insert your interrupt service routine here
```

```
;*****
```

```
*****
```

```
; Main
```

```
main:
```

```
    ; initialize stack pointer
```

```
    ldi temp, lo8(RAMEND)
```


```
    out SPL, temp
```

```
    ldi temp, hi8(RAMEND)
```

```
    out SPH, temp
```

```
    ; do your initializations here
```

```
*****
```





```
*****
```

```
infinite_loop:  
    ; nothing to implement here (interrupt driven program)  
    ; do the loop  
    rjmp infinite_loop
```