Report

Laboration 1

How to use the PORTs, Digital input/output and Subroutine call.

Author: Amata Anantaprayoon,
Adell Tatrous
Student ID: aa224iu,
at222ux
Semester: VT2018
Course: Computer Technology 1
Course code: 1DT301

Contents

1   Task 1: Light LED2

Write a program in Assembly language to light LED 2. You can use any of the four ports, but start with PORTB. The program should be very short! How many instructions is minimum number?

1.1   Flowchart
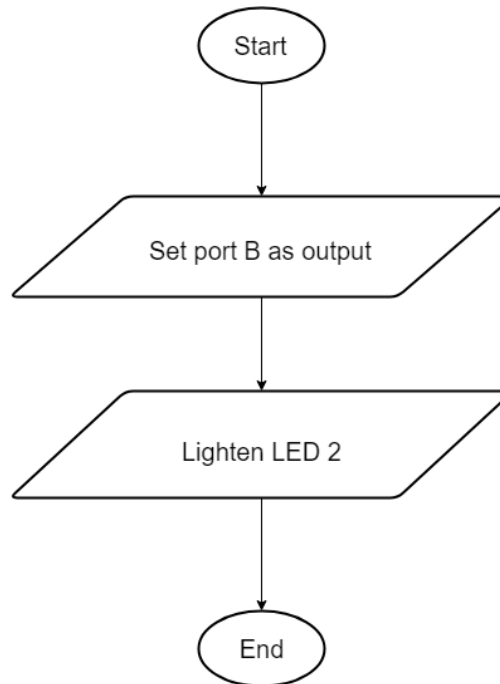
Figure 1: Flowchart for task 1

1.2   Assembly Program

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2018-09-13
; Author:
; Amata Anantaprayoon (aa224iu)
; Adell Tatrous (at222ux)
;
; Lab number: 1
; Title: How to use the PORTs. Digital input/output. Subroutine call.
;
; Hardware: STK600, CPU ATmega2560
;
; Function: Lights LED2 on PORTB
;
; Input ports: NONE
;
; Output ports: PORTB.
;
; Subroutines: N/A
; Included files: m2560def.inc
;
; Other information:
;
; Changes in program: 2018-09-13: Implementation
;                                 2018-09-14: Improve the code
;                         2018-09-16: Edit comments
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

.include "m2560def.inc"
.def dataDir = r16
.def ledOn = r17

;set portB to output
ldi dataDir, 0xFF  ;0b1111_1111
out DDRB, dataDir

;light LED2 (0 as on, 1 as off)
ldi dataDir, 0xFB ; 0b1111_10111
out PORTB, dataDir
```

2   Task 2: Switches And Light Corresponding LED.

Write a program in Assembly language to read the switches and light the corresponding LED. Example: When you press SW5, LED5 so should light. Make an initialization part of the program and after that an infinite loop.

2.1   Flowchart
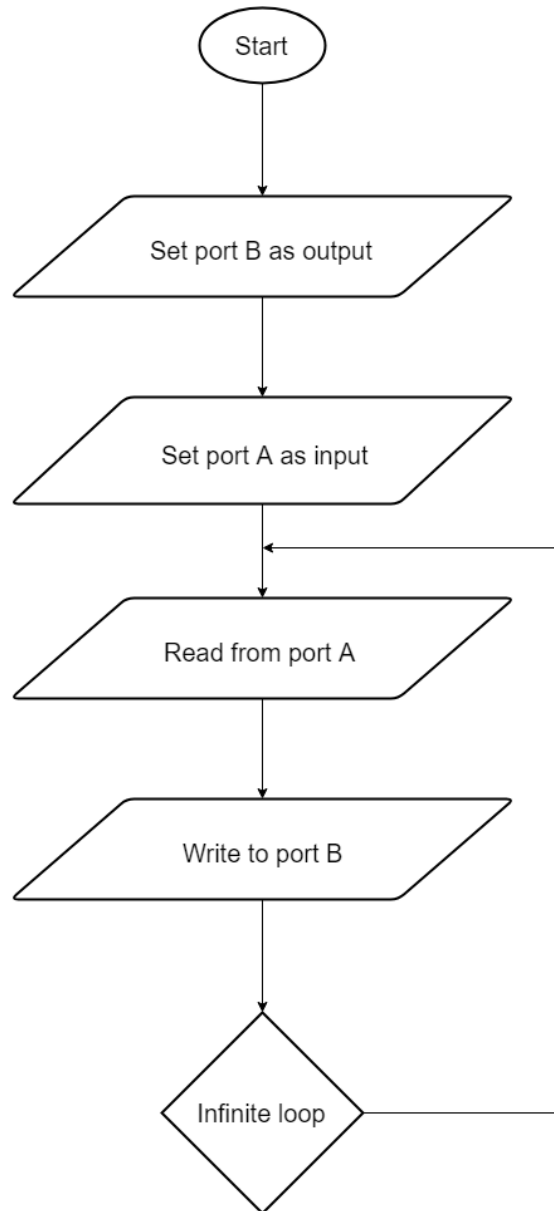
Figure 2: Flowchart for task 2

2.2 Assembly Program

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2018-09-13
; Author:
; Amata Anantaprayoon (aa224iu)
; Adell Tatrous (at222ux)
;
; Lab number: 1
; Title: How to use the PORTs. Digital input/output. Subroutine call.
;
; Hardware: STK600, CPU ATmega2560
;
; Function: Lights up LEDX when you press SWX. For instance, SW2 -> Lights LED2
;
; Input ports: On-board Switches are connected to PORTA
;
; Output ports: PORTB.
;
; Subroutines: N/A
; Included files: m2560def.inc
;
; Other information:
;
; Changes in program: 2018-09-13: Implementation
;                                 2018-09-14: Add loop
;                          2018-09-16: Edit comments
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<

.include "m2560def.inc"
.def dataDir = r16
.def switch = r17

; set portB as output
ldi dataDir, 0xFF
out DDRB, dataDir

;set portA as input
ldi dataDir, 0x00
out DDRA, dataDir

loop:

in switch, PINA              ;Read the content of PINA (switches) and safe it in "switch" (r17)
out PORTB, switch            ;Lights LEDX

rjmp loop
```

3 Task 3: SW5 Lights LED0

Write a program in Assembly language to read the switches and light LED0 when you press SW5. For all other switches there should be no activity.

3.1 Flowchart
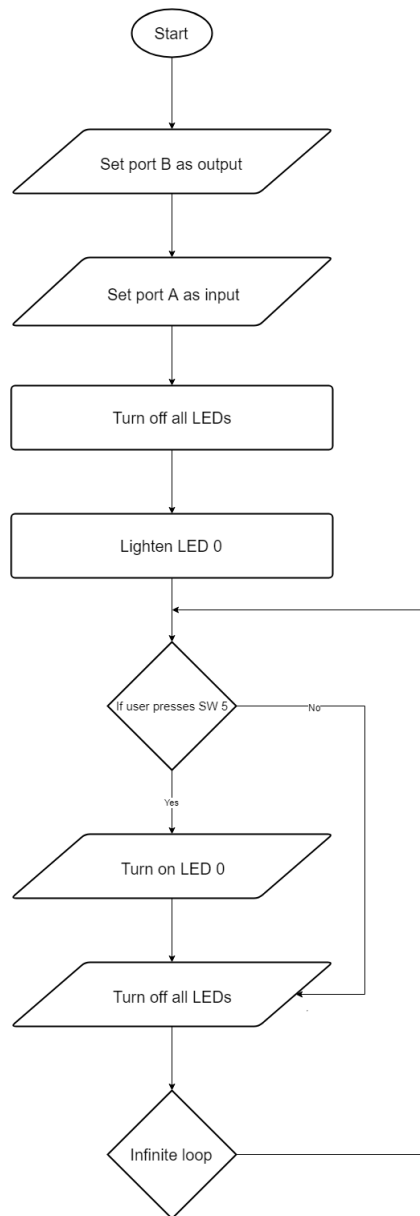
Figure 3: Flowchart for task 3

3.2 Assembly Program

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2018-09-13
; Author:
; Amata Anantaprayoon (aa224iu)
; Adell Tatrous (at222ux)
;
; Lab number: 1
; Title: How to use the PORTs. Digital input/output. Subroutine call.
;
; Hardware: STK600, CPU ATmega2560
;
; Function: Lights up LED0 when yoy press SW5
; Input ports: On-board Switches are connected to PORTA
;
; Output ports: PORTB.
;
; Subroutines: N/A
; Included files: m2560def.inc
;
; Other information:
;
; Changes in program: 2018-09-14: Implementation
;                     2018-09-16: Edit comments
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<



.include "m2560def.inc"
.def dataDir = r16
.def ledZero = r17  ; LED0
.def default = r18

;set port B as output
ldi dataDir, 0xFF
out DDRB, dataDir

;set port A as input
ldi dataDir, 0x00
out DDRA, dataDir


ldi default, 0xFF                       ;turn off all LED
ldi ledZero, 0b1111_1110        ;light only LED0

loop:
sbis PINA, PINA5                ; skip next line (Line 49) when user dont press SW5
out PORTB, ledZero             ; turn on LED0
out PORTB, default             ; default setting (LEDX = off)
rjmp loop
```

7

4 Task 5: Ring Counter

Write a program in Assembly language that creates a Ring Counter. The values should be displayed with the LEDs. Use shift instructions, LSL or LSR. Make a delay of approximately 0.5 sec in between each count. Write the delay as a subroutine. For using the subroutine, you must initialize the Stack Pointer, SP. Include the following instructions in beginning of your program: ; Initialize SP, Stack Pointer ldi r20, HIGH(RAMEND) ; R20 = high part of RAMEND address out SPH,R20 ; SPH = high part of RAMEND address ldi R20, low(RAMEND) ; R20 = low part of RAMEND address out SPL,R20 ; SPL = low part of RAMEND address

Function, the 8 LEDs: (0000 000X, 0000 00X0, 0000 0X00, 0000 X000, 000X 0000, 00X0 0000, 0X00 0000, X000 0000)

4.1 Flowchart
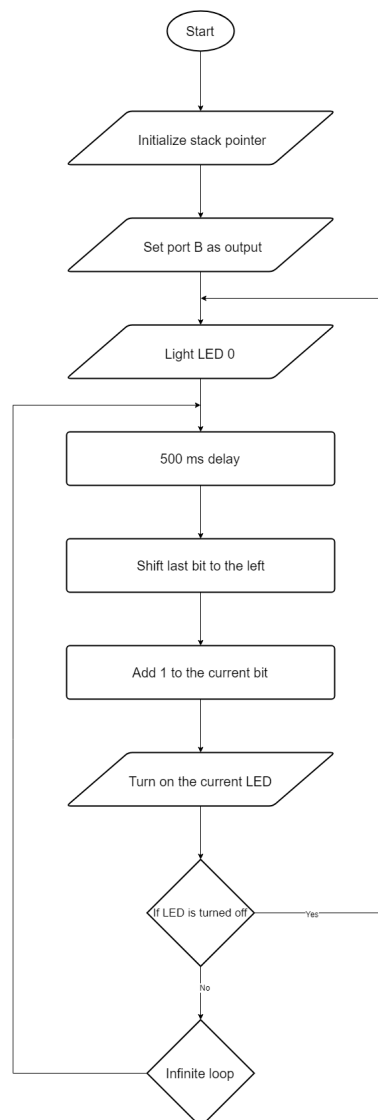
Figure 4: Flowchart for task 5

4.2 Assembly Program

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2018-09-14
; Author:
; Amata Anantaprayoon (aa224iu)
; Adell Tatrous (at222ux)
;
; Lab number: 1
; Title: How to use the PORTs. Digital input/output. Subroutine call.
;
; Hardware: STK600, CPU ATmega2560
;
; Function: Creats a Ring Counter
; Input ports: N/A
;
; Output ports: PORTB.
;
; Subroutines: 0.5 sec dealy in between each count
; Included files: m2560def.inc
;
; Other information: ; Generated by delay loop calculator
;                      at http://www.bretmulvey.com/avrdelay.html
;
; Changes in program: 2018-09-14: Implementation
;                     2018-09-16: Edit comments
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
  .include "m2560def.inc"
  .def dataDir = r16
  .def ledOn = r17

  ; Initialize SP, Stack Pointer
  ldi r20, HIGH(RAMEND)              ; R20 = high part of RAMEND address
  out SPH,R20                        ; SPH = high part of RAMEND address
  ldi R20, low(RAMEND)              ; R20 = low part of RAMEND address
  out SPL,R20                        ; SPL= low part of RAMEND address

  ;set port B as output
  ldi dataDir, 0xFF
  out DDRB, dataDir

start:

  ;Lights LED0
  ldi ledOn, 0b1111_1110
  out PORTB, ledOn

loop:

; Delay 500 000 cycles
; 500ms at 1 MHz

    ldi  r18, 3
    ldi  r19, 138
    ldi  r20, 86
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
    rjmp PC+1

lsl ledOn                         ; shift last bit to the left
inc ledOn                         ; add 1 to the current bit
out PORTB, ledOn

cpi ledOn, 0xFF                    ; compare ledOn with 0xFF
breq start                        ; IF ledOn = 0xFF jump to start

rjmp loop
```

9

5   Task 6: Johnson Counter

Write a program in Assembly language that creates a Johnson Counter in an infinite loop.

Function, the 8 LEDs: (0000 000X, 0000 00XX, 0000 0XXX, 0000 XXXX, 000X XXXX, 00XX XXXX, 0XXX XXXX, XXXX XXXX, 0XXX XXXX, 00XX XXXX, 000X XXXX, 0000 XXXX, 0000 0XXX, 0000 00XX, 0000 000X, 0000 0000)
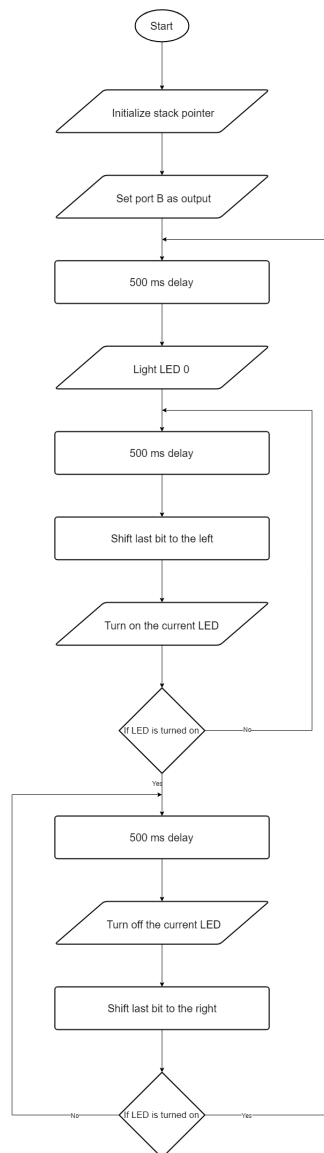
5.1   Flowchart

Figure 5: Flowchart for task 6

5.2   Assembly Program

```
;>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
; 1DT301, Computer Technology I
; Date: 2018-09-14
; Author:
; Amata Anantaprayoon (aa224iu)
; Adell Tatrous (at222ux)
;
; Lab number: 1
; Title: How to use the PORTs. Digital input/output. Subroutine call.
;
; Hardware: STK600, CPU ATmega2560
;
; Function: creates a Johnson Counter in an infinite loop
; Input ports: N/A
;
; Output ports: PORTB.
;
; Subroutines:
; Included files: m2560def.inc
;
; Other information:  Generated by delay loop calculator
; at http://www.bretmulvey.com/avrdelay.html
;
; Changes in program: 2018-09-14: Implementation
;                     2018-09-16: Edit comments
;
;<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<


.include "m2560def.inc"
.def dataDir = r16
.def ledOn = r17
.def ledOff = r22


; Initialize SP, Stack Pointer
ldi r20, HIGH(RAMEND)                    ; R20 = high part of RAMEND address
out SPH,R20                              ; SPH = high part of RAMEND address
ldi R20, low(RAMEND)                     ; R20 = low part of RAMEND address
out SPL,R20=                             ; SPL= low part of RAMEND address

;set port B as output
ldi dataDir, 0xFF
out DDRB, dataDir
    start:

    ; Delay 500 000 cycles
    ; 500ms at 1 MHz
        ldi  r18, 3
        ldi  r19, 138
        ldi  r20, 86
    L1: dec  r20
        brne L1
        dec  r19
        brne L1
        dec  r18
        brne L1
        rjmp PC+1

    ;Lights LED0
    ldi ledOn, 0b1111_1110
    out PORTB, ledOn

    ;Load LedOff for backward "loop"
    ldi ledOff, 0b0111_1111


    forward:

        ldi  r18, 3
        ldi  r19, 138
        ldi  r20, 86
    L2: dec  r20
        brne L2
        dec  r19
        brne L2
        dec  r18
        brne L2
        rjmp PC+1

    lsl ledOn                       ; shifts last bit to the left
    out PORTB, ledOn
    cpi ledOn, 0x00                 ; compare ledOn with 0xFF
    breq backward                   ; IF ledOn = 0xFF jump to start

    rjmp forward

    backward:

        ldi  r18, 3
        ldi  r19, 138
        ldi  r20, 86
    L3: dec  r20
        brne L3
        dec  r19
        brne L3
        dec  r18
        brne L3
        rjmp PC+1

    out PORTB, ledOff

    lsr ledOff                      ; shifts first bit to the right
    cpi ledOff, 0x00                ; compare ledOff with 0x00
    breq start                      ; IF ledOff = 0x00 jump to start

    rjmp backward
```