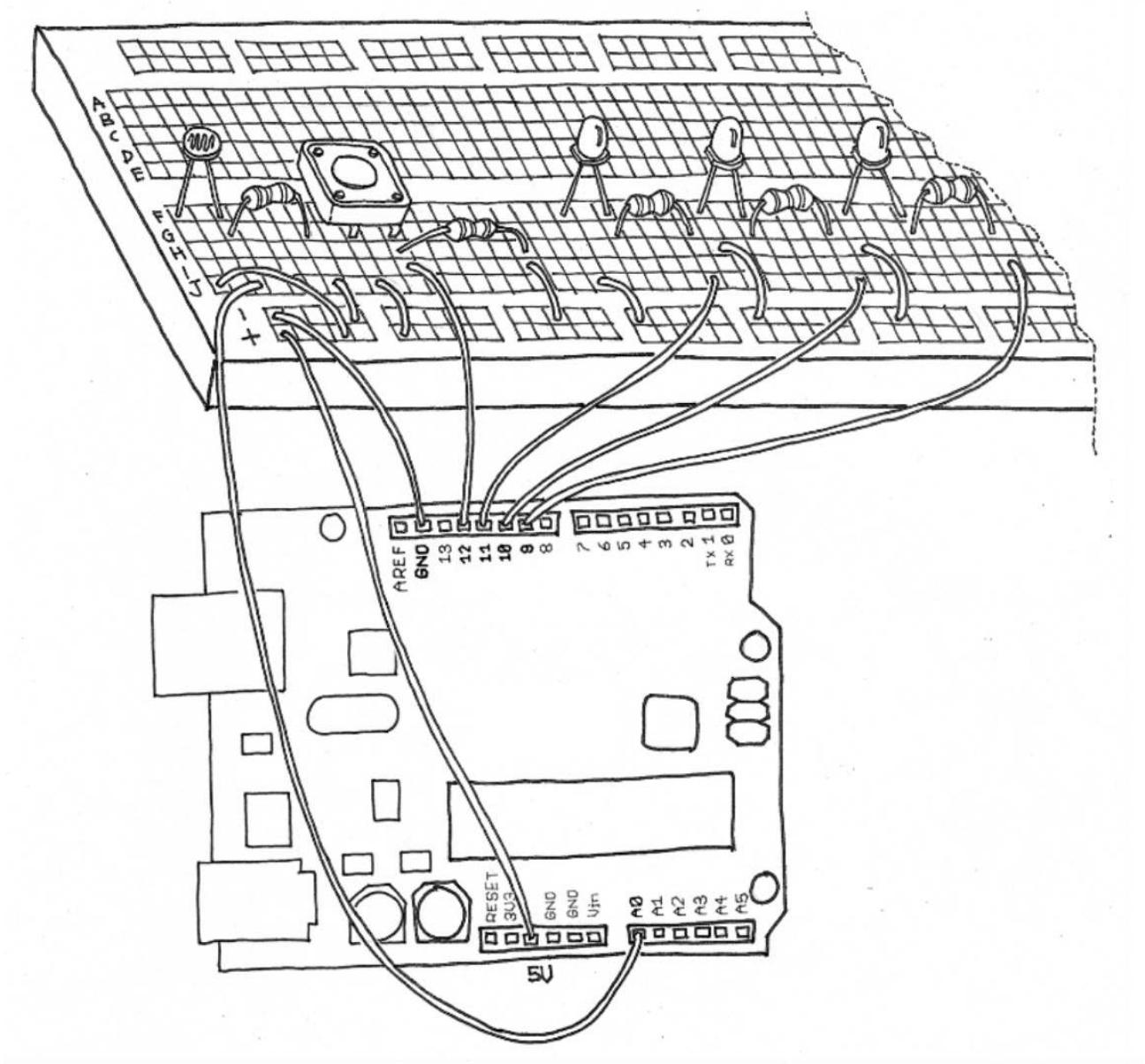
CS/EEE/INSTR F241
Microprocessor Programming and Interfacing

Lab 7 - Advanced Operations using Interrupts



Dr. Vinay Chamola and Anubhav Elhence

A complicated Example:

Write an ALP that does the following:

(1) Display the string “Enter 10 character long User Name” and goes to the next line

(2) Takes in the user-entered string of 10 characters and compares with the user name value already stored in memory

(3) If there is no match it should exit saying “wrong Username”

(4) If there is a match it should display the string “Enter 5 character long Password” and goes to the next line

(5) Takes in the password entered by the user and compares it with the password already stored in memory

(6) If there is no match it should exit’

(7) If there is a match it should display “Hello <Username>” where <Username> is replaced by the actual username of the person.

Sample Output:

```
-g 01af
enter 10 character long User Name:
anub@g.com
enter 5 character long password:
******
hello anub@g.com
AX=092A BX=0000 CX=0000 DX=0272 SP=FFFE BP=0000 SI=0230 DI=0235
DS=0863 ES=0863 SS=0863 CS=0863 IP=01AF NV UP EI PL NZ NA PO NC
0863:01AF 8D1E0902          LEA     BX,[0209]                   DS:0209=6E65
```

```
; This Assembly Language Program (ALP) checks the entered username and password, and displays a custom message accordingly.

.model tiny

.data

; Data section contains the messages, the correct username, and password for comparison.

msg1 db "enter 10 character long User Name: $" ; Message 1: Prompt to enter the username

usn1 db "anub@g.com$" ; Correct username for comparison

max1 db 20 ; Maximum length for input

act1 db ? ; Placeholder for action

inp1 db 20 dup("$") ; Buffer to store user's input for username

msg2 db "enter 5 character long password:$" ; Message 2: Prompt to enter the password
```

```
pass1 db "oscar" ; Correct password for comparison

inp2 db 30 dup("$") ; Buffer to store user's input for password

msg3 db "hello $" ; Message 3: Greeting message when both inputs are correct

msg4 db "wrong username$" ; Message 4: Wrong username input

msg5 db "wrong password$" ; Message 5: Wrong password input

nline db 0ah, 0dh, "$" ; New line characters

.code

.startup

; Display message 1 on the screen and go to the next line.

    lea dx, msg1

    mov ah, 09h

    int 21h

; Add a new line after the message.

    lea dx, nline
```

```
    mov ah, 09h
    int 21h
; Take input from the user and store it in inp1.
    lea dx, max1
    mov ah, 0ah
    int 21h
; Compare the entered username with the stored username.
 cld
    lea di, inp1
    lea si, usn1
    mov cx, 11
 repe cmpsb
    jcxz l1
; If the username is incorrect, display the "wrong username" message and exit.
    lea dx, nline
```

```
    mov ah, 09h
    int 21h
    lea dx, msg4
    mov ah, 09h
    int 21h
    mov ah, 4ch
    int 21h
; If the username is correct, display the "enter password" message.
    l1:
    lea dx, nline
    mov ah, 09h
    int 21h
    lea dx, msg2
    mov ah, 09h
    int 21h
```

```
    lea dx, nline

    mov ah, 09h

    int 21h

; Take password input from the user, masking the characters.

    mov cx, 6

    lea di, inp2

    l2:

    mov ah, 08h

    int 21h

    mov [di], al

    mov dl, '*'

    mov ah, 02h

    int 21h

    inc di

    dec cx

    jnz l2
```

```
; Compare the entered password with the stored password.

 cld

    mov cx, 6

    lea di, inp2

    lea si, pass1

 repe cmpsb

    jcxz l3

; If the password is incorrect, display the "wrong password" message and exit.

    lea dx, nline

    mov ah, 09h

    int 21h

    lea dx, msg5

    mov ah, 09h

    int 21h

    mov ah, 4ch
```

```
    int 21h

; If the password is correct, display the greeting message and
the username.

    l3:

    lea dx, nline

    mov ah, 09h

    int 21h

    lea dx, msg3

    mov ah, 09h

    int 21h

    lea dx, usn1

    mov ah,09h

    int 21h

    lea dx, nline

    mov ah, 09h

    int 21h
```

```
; lea bx, msg2

; lea bx, msg2

; lea bx, msg2

; lea bx, msg2

.exit

end
```

Lab Task

Task 1: Take the above problem and modify the ALP such that instead of taking username and password in the .data section, we take it from the user.txt and pswd.txt file.

The user.txt file contains 10 character long username with 11th character being '$' and pswd.txt file contains 5 character long password with 6th character being '$'.

Task 2: Now modify the ALP in Task 1, such that the username and password can be of variable length with max number of characters being 20h (i.e. 32 in decimal). The username and password have to be picked from user.txt and pswd.txt file respectively.


BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
ज्ञानं परमं बलम्