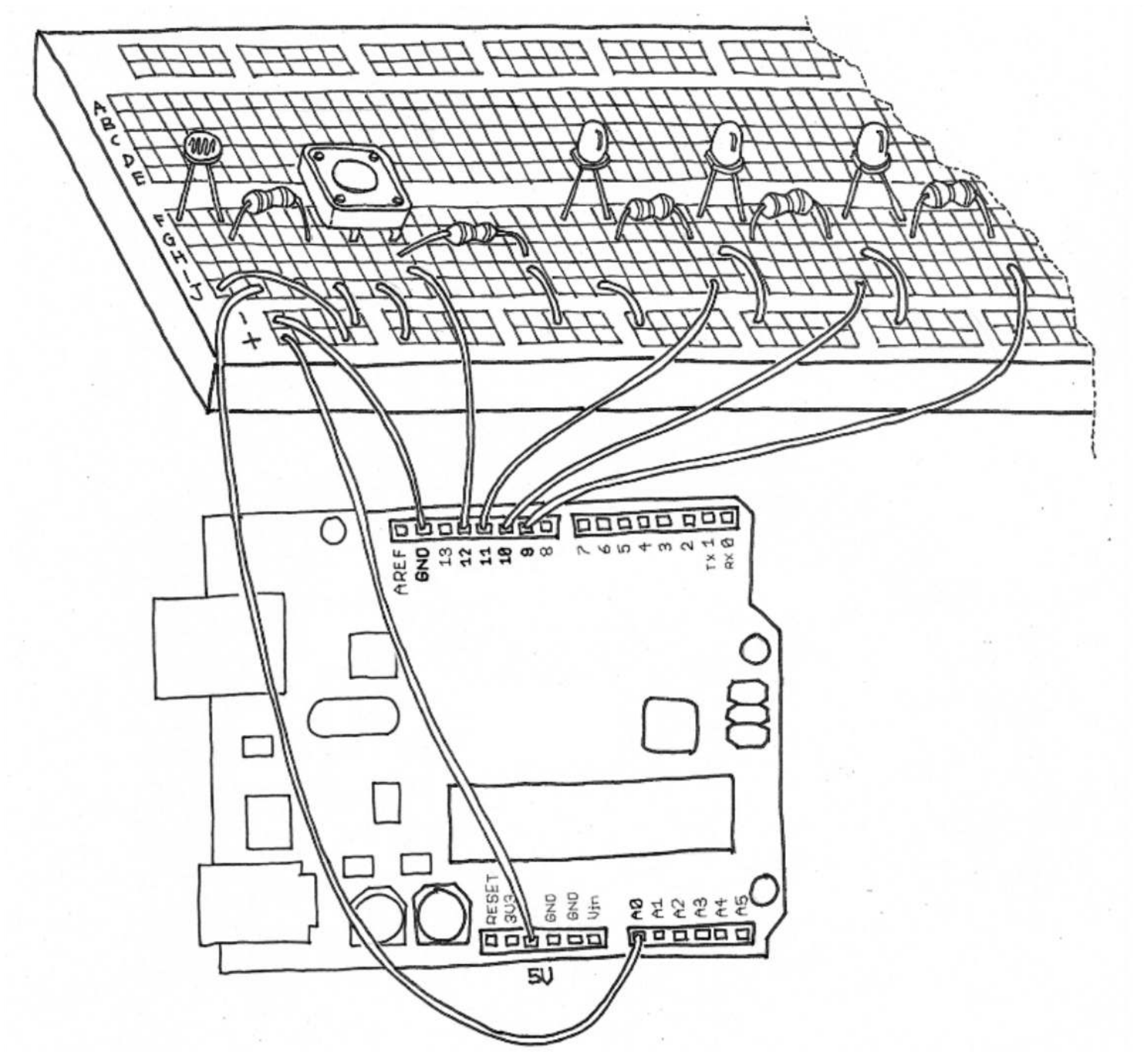
CS/EEE/INSTR F241
Microprocessor Programming and Interfacing

Lab 5 - Dealing with Interrupts



Dr. Vinay Chamola and Anubhav Elhence

Introduction to DOS & BIOS Interrupts

The BIOS is specific to the individual computer system. It contains the default resident hardware-dependent drivers for several devices including the console display, keyboard, and boot disk device. Have you ever wondered how you can type characters and view them even when no OS is loaded, like when the computer is booting?

The BIOS operates at a more primitive level than the DOS kernel. In X86-based systems, there are a total of 256 possible interrupts, out of which some interrupts are reserved for DOS, and BIOS services, which the programmer can access by loading the function number into appropriate registers and then invoking the interrupt.

In providing the DOS functions, the MS-DOS kernel communicates with the BIOS device drivers through the BIOS functions. DOS then provides something like a wrapper over the primitive BIOS function. By providing the wrapper, DOS makes the functions easier to use and more reliable, but in turn, reduces the flexibility and power of the functions. By using the BIOS interrupts directly, the programmer bypasses the DOS kernel.

In the following two experiments, you will be using INT 21 H (DOS Interrupts) – Here we use INT 21H to access the value of the key pressed and display characters.

Keyboard Interrupts

Purpose: Input a character from the keyboard (STDIN) with Echo

```
MOV AH, 01h                                ; AH -01 parameter for INT 21h
INT 21h
```

You have to run the program using debug/debug32 and the ASCII key you press will be stored in AL register.

Purpose: Input a character from the keyboard (STDIN) without Echo

```
MOV AH, 08h                                ; AH -08 parameter for INT 21h
INT 21h
```

You have to run the program using debug/debug32 and the ASCII key you press will be stored in AL register. The difference is the character will not be visible on the screen when you type it

Purpose: Input a string from keyboard (STDIN)

```
.data
max1 db 32                    ; 32 is max no. of chars that a user can type in (max possible – 255)
Act1 db ?                     ; actual count of keys that user types will be stored here after int has
                              ; executed (Note this cannot exceed the value specified in max1 –
                              ; actual keys you enter will 31 as the 32nd will be Enter key)
Inp1 db 32dup(0)              ; Reserve 32 locations for input string

.code
.startup

LEA DX,max1
MOV AH, 0Ah
INT 21h
```

After the interrupt, act 1 will contain the number of characters read, and the characters themselves will start at inp1 The characters will be terminated by a carriage return (ASCII code 0Dh), although this will not be included in the count (Note: this will not be included in the ACT1 but you have to count Enter also when you are specifying it in max1)

Purpose: Output a character to display (STD OUT)

```
MOV   DL, 'A'
MOV   AH, 02h
INT     21h
```

After Interrupt is executed character 'A' will be displayed on the screen.

Purpose: Output a string on display (STDOUT)

```
.data
str1 db 'HELLO$'                    ; all strings must terminate with '$' ASCII value (24h)

.code
.startup

LEA DX, str1
MOV AH, 09h
INT 21h
```

When interrupt is executed the string "HELLO" will be displayed on screen. Remove the '$' sign. What happens

Lab Task 1

Write an ALP that will take in a alphabet entered by the user and display ‘The character entered is a’ if the character entered is ‘a’(in both cases) else display ‘not a’ if the character entered is any other character but ‘a’. The user entered character should not be seen on the screen.

Lab Task 2

Write an ALP to convert an input string from the user (in all lowercase letters only) to uppercase letters and display it to the user.

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

ज्ञानं परमं बलम्