

Hints for Lab 2

Notes

Project File

In addition to creating the LAB2.PRJ file, you need to tell Turbo C the name of the project file to use. Do this by using the Project→Project Name menu item and entering the name of the project file (e.g. LAB2.PRJ). You will need to do this every time you start Turbo C.

Simplification

You may write the `rdkbd()` function so that it returns a character value instead of a number.

Undefined Bits

Note that the upper four bits of the value read from the data port may not be equal to zero.

Avoid Repeating Numbers

Before trying to detect whether a button has been pushed you should wait until all buttons have been released. This will avoid having one button press detected several times.

Overall Organization

At the start of your program you should have the line:

```
#include <iolib.h>
```

This should be followed by the functions *you* write with a `main()` function at the end of the file. The `speek` and `spoke` functions will appear to be built-in functions (like `printf()`) — you do not have to declare or write them yourself.

Example Code

Writing to the Display

The following lines of code write the character given by the value of the variable `c` to the `i`'th LED from

the right:

```
spoke ( 208, c + 128 ) ;  
spoke ( 209, i ) ;  
spoke ( 209, i + 8 ) ;  
spoke ( 209, i ) ;
```

Shifting Characters

The following code shifts the characters in the 8-character array `x` right (towards higher indices) by 1 position and then stores the value of the variable `y` in the first position. The variables `i`, `x`, and `y` are assumed to have been declared elsewhere.

```
i = 7 ;  
while ( i > 0 ) {  
    x[i] = x[i-1] ;  
    i = i - 1 ;  
}  
x[0] = y ;
```

Testing for a Button Push

The following code waits for all buttons to be released:

```
spoke ( 208, 0xF ) ;  
while ( speek ( 208 ) & 0xF ) { } ;
```

The following code sets the third column of the keypad matrix to a logic '1' and then test whether the top-most row is a logic '1'. If so, then the E key has been pressed and the code sets the value of the variable `key` to 14.

```
spoke ( 208, 4 ) ;  
if ( speek ( 208 ) & 8 ) {  
    key = 14 ;  
}
```

Organization of `rdkbd`

The following code shows some of the statements that might be included in the `rdkbd()` function:

```
int rdkbd ( void )
{
    int key ;
    key = -1 ;

    <statements to wait until all buttons are released>

    while ( key == -1 ) {
        <statements to detect a button press and set "key">
    }

    return key ;
}
```

If you declare the following array:

```
int int2hex[16] = {
    '0', '1', '2', '3', '4', '5', '6', '7',
    '8', '9', 'A', 'B', 'C', 'D', 'E' } ;
```

Then the following expression converts the integer *i* (which must be between 0 and 15) to the equivalent hex character:

```
c = int2hex[i] ;
```