

Lab 4: Keypad

ECE 3720

Preview

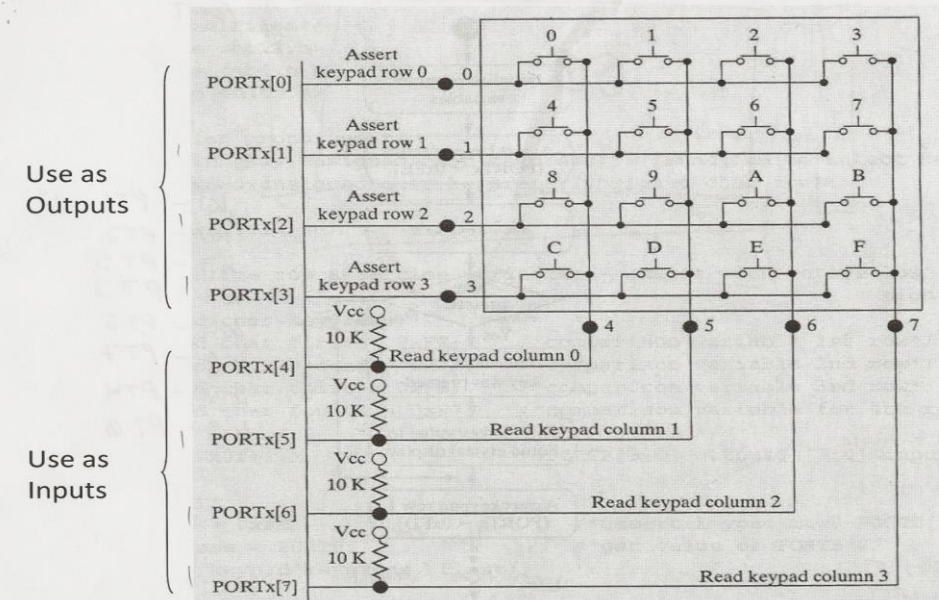
A keypad will be connected to the board and the output will be displayed as a binary on microcontroller output pins.

Topic	Slide
Keypad Overview	<u>3</u>
Lab Goals	<u>5</u>
Key Mask	<u>6</u>

Keypad Overview

- Each button on the keypad is a button just like the one used in Lab 2.
 - This necessitates the use of pulling resistors
- This shows one method of setting up a keypad
 - Each column is set as input and **pulled up** to digital high
 - Each row is set to digital low one at a time (the rest are set to high)
 - The inputs (columns) are scanned and if one reads low, that means the button corresponding to the row and column is pressed
- How could we change this?

Keypad Figure 5.5



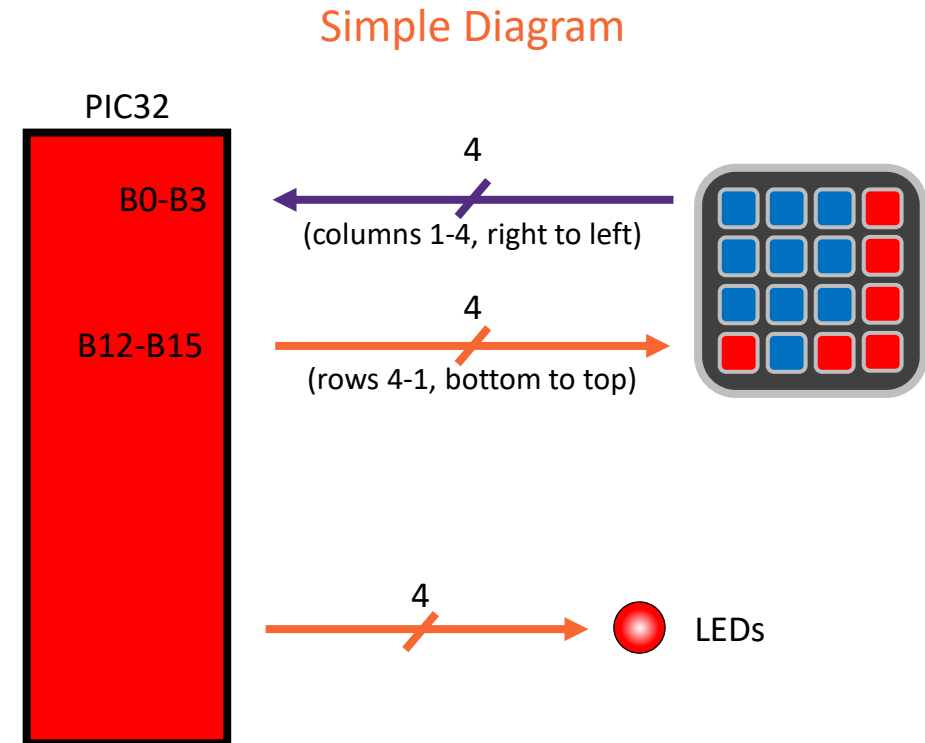
Keypad

- Datasheet available on Canvas and [here](#)
 - This will show you the pinouts for this specific keypad
 - See the *How it Works* section on page 2
- It is the user's responsibility to supply the necessary pulling resistors.



Lab Goals

- The keypad will be connected as shown to the right
 - You'll need to refer to the datasheet for the appropriate pins on the keypad
 - You will use internal pull-down resistors (taught in Lab 2) instead of pull-ups shown earlier
- 4 LEDs will be connected to separate outputs of the MC and indicate which key was pressed.
- When you push a key, the AD2 will display that key's value in binary via Static IO



Key Mask

- The **mask** assumes you connect the wires EXACTLY as described on the previous slide
- The purpose of the **mask** is to simplify the setting of each column bit and scanning the rows
 - Each value corresponds to a specific row/column combo that will only be true when a button is pushed
 - Think (for) loops
 - You may need to use bit-masking when scanning the inputs (*input&0xF*)
- key_mask** gives you the corresponding output for each key
 - Note that A-D, *, and # have been replaced with 10-15. We can't display letters and symbols on LEDs

```
unsigned int mask[16] = {0x8008, 0x8004, 0x8002, 0x8001,
                        0x4008, 0x4004, 0x4002, 0x4001,
                        0x2008, 0x2004, 0x2002, 0x2001,
                        0x1008, 0x1004, 0x1002, 0x1001};
unsigned char key_mask[16] = {1,  2,  3, 10,
                             4,  5,  6, 11,
                             7,  8,  9, 12,
                             14, 0, 15, 13};
```