Keypad decoding, 7-segment display

Overview

- Lab for Monday
  - In C, add numbers input on 4x4 keypad
  - Display result as a single hex digit (0-15)
  - Hardware:
    - Keypad
      - How to read key press?
      - How to handle switch bounce?
    - 7-segment display
      - Review from previous lab
      - How to convert to C?
      - Displaying 0-9 plus A, B, C, D, E, F

Keypad

- Keypads
  - A common input device for microcontrollers
  - Main problem:
    - 4 x 4 keypad = 16 input lines = too many
    - How to connect in an I/O line efficient manner?
  - Also need to debounce:
    - Single keypress = single event in your program

Reading a keypad

- 4x4 keypad
  - 8 output pins
  - Hook to a single port
  - We'll use P3

- Reading procedure:
  - Select a row/column setting a single pin low
  - Then read all 8 pins
    - Any pushed button will be low (0)
    - Multiple buttons could be pushed at same time
    - Only read key pushed in selected row/column

Keypad wiring

- Our 4x4 keypads:
  - Pins labeled 1-8
  - Hook pin 1 to port pin 0, pin 2 to port pin 1, ...
- Row selection:
  - 0 in low nibble determines which row
    - 1110 = selects row 123A
    - 1101 = selects row 456B
    - 1011 = selects row 789C
    - 0111 = selects row *0#D

Reading keypad

- Column selection:
  - 0 in high nibble determines column
    - 1110 = column 147*
    - 1101 = column 2480
    - 1011 = column 369#
    - 0111 = column abcd

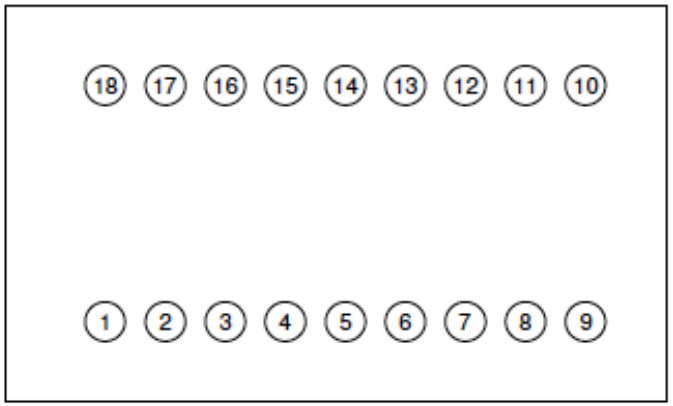
Reading procedure

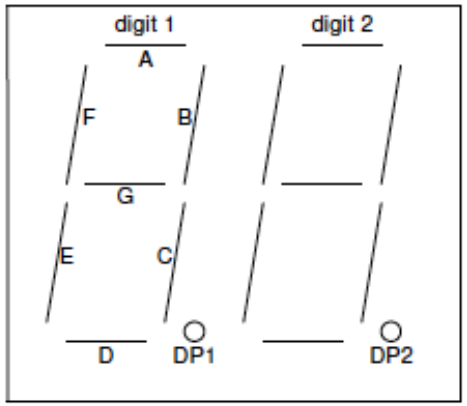
- Scan through 4 rows or columns
  - Except for row/column selection bit, all other pins high to allow subsequent reading
  - Copying row/column select byte to port
  - Read byte from port
    - Other nibble will have bits low if that button pushed
    - In row selection, leftmost button is least significant position in read nibble
  - Example:
    - Select byte: 1111 1110 = selects row 123A
    - Result byte: 1101 1110 = key 2 down

7-segment display

- Using two digit display, only using first digit
  - Use a ~330 Ohm resistor
- Bring pin low to light a segment
  - 8052 serves as the current sink
- Determine the byte pattern to send to port
  - Single copy of a byte can set all 7-segments
- Lowest pin connect to lowest segment letter
  - e.g. P2.0 = segment A, P2.1 = segment B, ...

7-segment details





digit 1 segment E	1	18	digit 1 segment F
digit 1 segment D	2	17	digit 1 segment G
digit 1 segment C	3	16	digit 1 segment A
DP1	4	15	digit 1 segment B
digit 2 segment E	5	14	digit 1 common (+)
digit 2 segment D	6	13	digit 2 common (+)
digit 2 segment G	7	12	digit 2 segment F
digit 2 segment C	8	11	digit 2 segment A
DP2	9	10	digit 2 segment B

Summary

- Keypad decoding
  - Multi-step process to determine which key(s) are pressed
  - Necessary because in general we can't afford an I/O line per key
- 7-segment display
  - Convert assembly to C code
  - Add support for hex digits A-F