

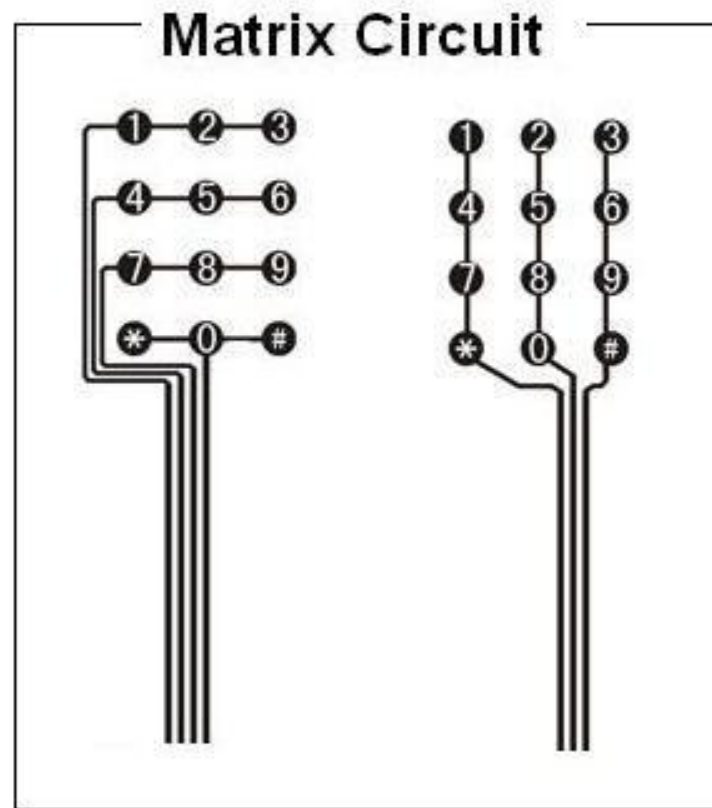
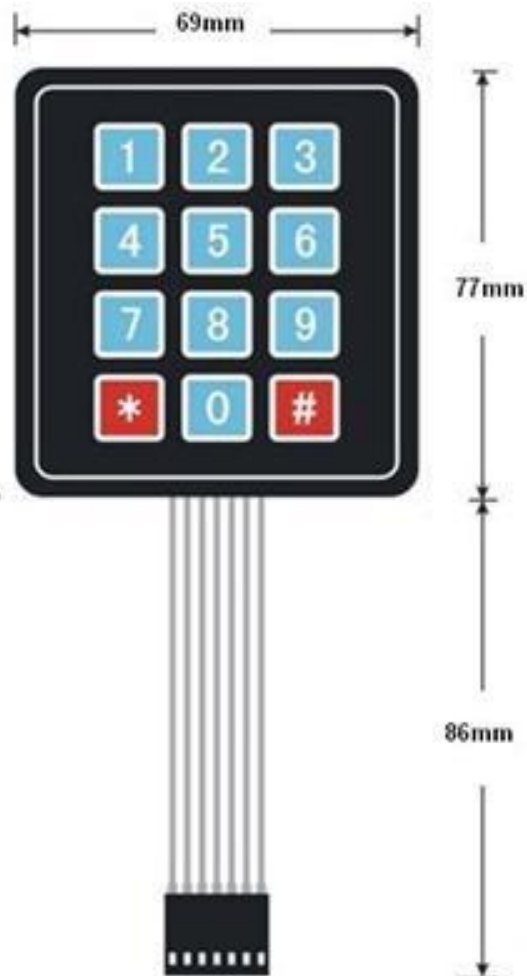
Exercise 8 Prep

External Interrupts and Keypad

Main Points

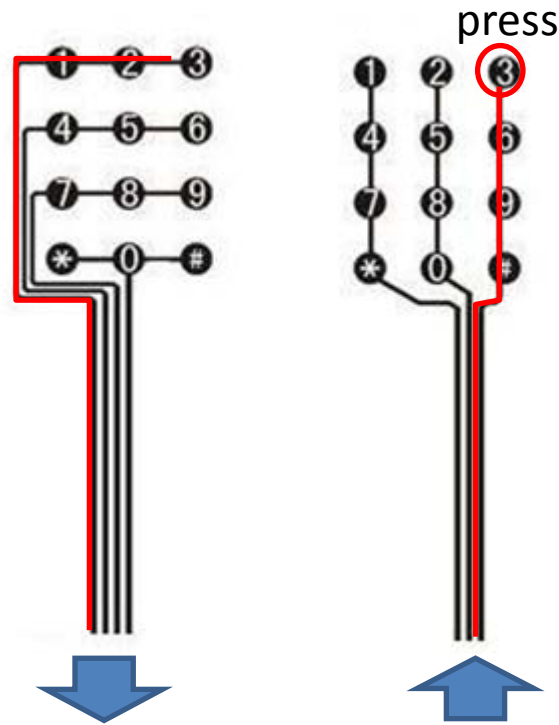
- How external interrupts work HW/SW
- Tips for all interrupt service routines (keep them small, no waiting, etc)
- How keyboards work

Matrix Keypad



Pressing a key connects one of the row wires to one of the column wires

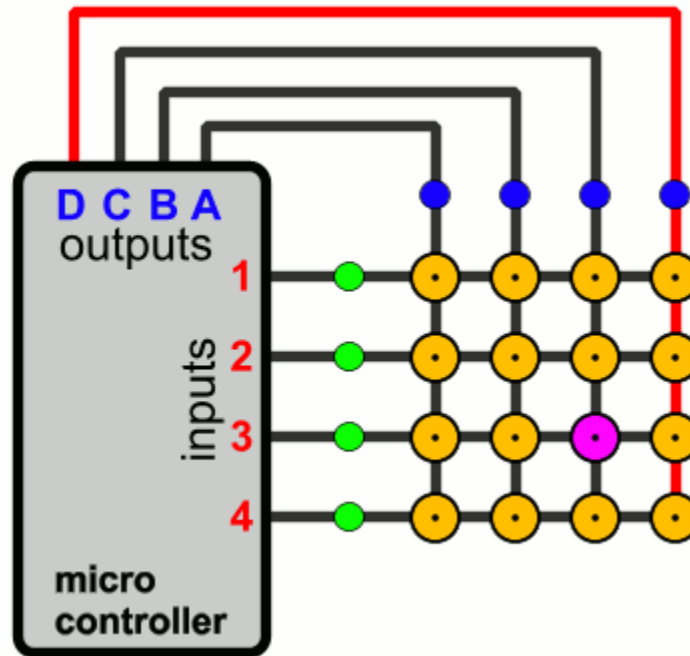
Interfacing the Keypad



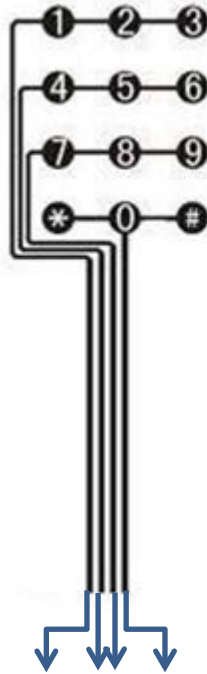
Detect an “active” signal on one row if a key is pressed.

Scan the columns by cycling one “active” signal across all columns and repeat.

Animation – 4x4 Example



PIC24 Strategy



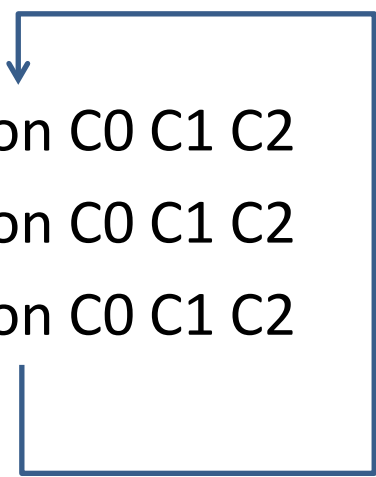
Connect to CN pins
Enable internal pullups (CNPU1)
Enable CN interrupts (CNEN1)



Connect to digital output pins
Poll with one output active
LOW at a time.

Software Strategy

In Main Loop:



Send 011 on C0 C1 C2
Send 101 on C0 C1 C2
Send 110 on C0 C1 C2

In `_CNInterrupt()` ISR:

- Save status of column
- Wait for switch bounce (25ms)
- If a key is pressed (remember that CN interrupt will occur on falling OR rising edge of input)
 - Set the row status
- Wait for key to be released
- Service the (row, col) key

Change Notification Registers and Bits

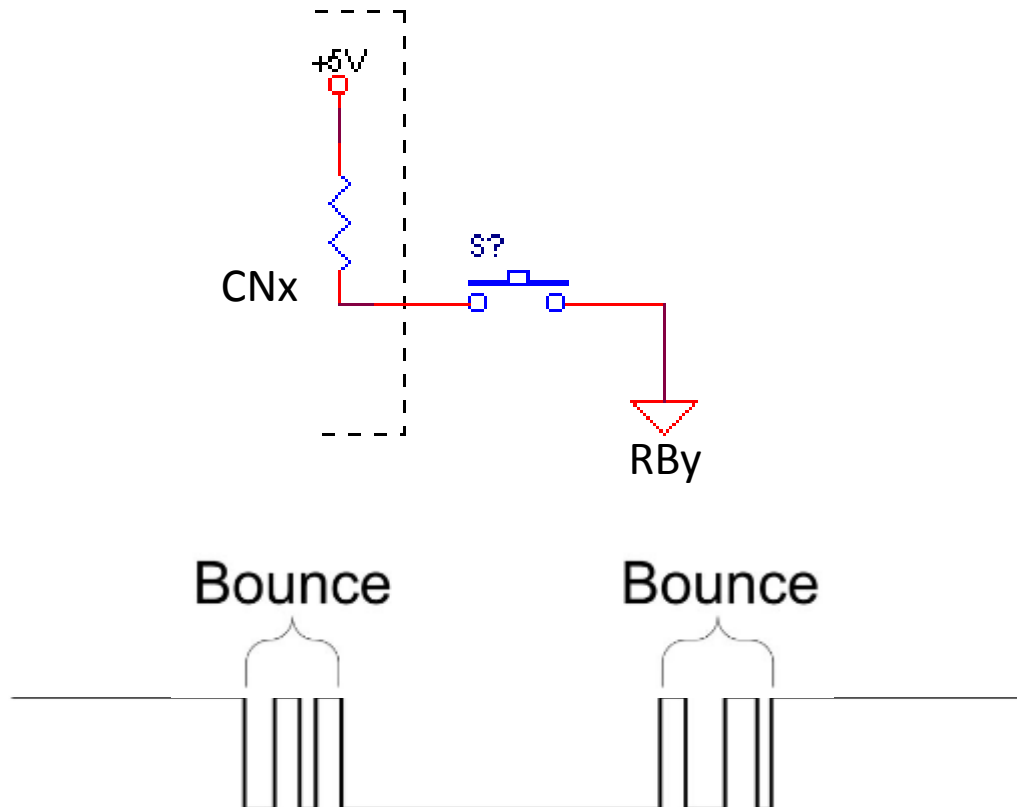
- `_CNIF` – Change Notification Interrupt Flag
- `_CNIE` – Change Notification Interrupt Enable
- `CNPU1`, `CNPU2` – for enabling internal pullup
- `CNEN1`, `CNEN2` – for enabling interrupts
- Library functions defined in `pic24_ports_config.h`

For more information see section 11.5 in PIC24 Datasheet

Change Notification Interrupts

- Any change on an enabled CNx pin causes a CN Interrupt
 - Rising edge
 - Falling edge
- If more than one CNx pin is enabled, the `_CNInterrupt()` interrupt service routine executes if ANY of the pins change.
- `_CNInterrupt()` ISR must determine which CN pin caused the interrupt.

Interrupting with Switches



ISR must handle the bounce

ISR can occur on both button press and release

Decoding the Key with Matrix

- In C, a matrix is a 2-dimensional array

- Example:

```
int disp[2][3]; // A 2 x 3 array
```

- Initializing – **must always specify size:**

```
int disp[2][4] = { {10, 11, 12, 13},
                  {14, 15, 16, 17} };
```

or

```
int disp[2][4] = { 10, 11, 12, 13, 14, 15, 16, 17};
```

- To specify an array element:

```
int_x = disp[0][3]; //int_x = 13
```