Lab 3: matrix Keyboard and interrupt

In this lab, the main goals are

1. To get familiar with the keypad operation
2. To get familiar with Arduino libraries
3. To get familiar with interrupt operation

Important Knowledge

attachInterrupt(), detachInterrupt(), digitalPinToInterrupt()

interrupts(), noInterrupts()

millis(), micros(), delay(), delayMicroseconds()

The digitalPinToInterrupt() function is not documented in Arduino's reference page, here is the link from the Arduino's github issue tracking page: https://github.com/arduino/Arduino/issues/2676

Please pay attention to what pins are usable for interrupts!

Board	int.0	int.1	int.2	int.3	int.4	int.5
Uno, Ethernet	2	3				
Mega2560	2	3	21	20	19	18
32u4 based (e.g Leonardo, Micro)	3	2	0	1	7	
Due, Zero			(see below)			

Board:	Digital Pins Usable For Interrupts
Uno, Nano, Mini, other 328-based	2, 3
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21
Micro, Leonardo, other 32u4-based	0, 1, 2, 3, 7
Zero	all digital pins, except 4
Due	all digital pins

Please make sure you create one sketch for each exercise and copy the code to the report! Do not submit your code directly, the submission only allows doc, docx or pdf.

Exercises

1. **Preparation (15 min): Connect** the keypad to Arduino and **install** the keypad library if it is not installed. You need to check this page at first: http://playground.arduino.cc/Code/Keypad. No coding for this exercise.
2. **Easy (5 min):** Whenever a key is pressed, print the key value to the host PC through serial communication.
3. **Difficult (30 min):** Implement a simple calculator which can only do multiplication
    a. Use the "#" key as "=", use the "*" key as multiplication
    b. If multiple keys are pressed before "*" or "*", those keys form an integer
    c. if "#" is pressed, print the equation to the host PC, e.g., 11*12*13=1716
4. **Difficult (60+ min):** Implement a simple alarm system. You may use the keypad.addEventListener() method or other methods provided in the keypad library. You are welcome to develop your own methods too.
    a. A simple "Alarm" button.
        i. When "#" has been hold for over 2 seconds, the onboard LED will be turned on.
        ii. When "*" is pressed, the onboard LED with be turned off.
        iii. The LED should not be turned off before "*" is pressed.

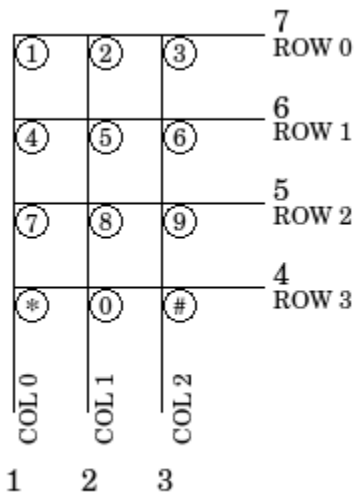
b. An advanced alarm system
    i. If “#” and “*” are pressed at the same time, keep sending alarm messages to the host PC, and blinking the onboard LED, until both “#” and “*” are released.

5. **Ninja (60+ min):** Implement your own keypad scan method using interrupt (event driven) instead of scanning (polling).
    a. The actual implementation of keypad library uses a scan method as follows

```
// Private : Hardware scan
void Keypad::scanKeys() {
    // Re-intialize the row pins. Allows sharing these pins with other hardware.
    for (byte r=0; r<sizeKpd.rows; r++) {
        pin_mode(rowPins[r],INPUT_PULLUP);
    }

    // bitMap stores ALL the keys that are being pressed.
    for (byte c=0; c<sizeKpd.columns; c++) {
        pin_mode(columnPins[c],OUTPUT);
        pin_write(columnPins[c], LOW);  // Begin column pulse output.
        for (byte r=0; r<sizeKpd.rows; r++) {
            bitWrite(bitMap[r], c, !pin_read(rowPins[r]));  // keypress is active low so invert to high.
        }
        // Set pin to high impedance input. Effectively ends column pulse.
        pin_write(columnPins[c],HIGH);
        pin_mode(columnPins[c],INPUT);
    }
}
```

    b. Your code should catch the interrupt generated by the keypad column and then decode the actual key.



    c. Print each pressed key value to the host in your main program, e.g., loop()

What to submit

1. Your lab report, please use the template in canvas, and submit in word or pdf.
2. Please do follow the lab report template and don’t miss any required content.
3. Please be sure to copy all your code to the report and follow the requirement.