ECE 367 - Experiment #3
Matrix Keypad Interfacing

Spring 2006 Semester

Introduction

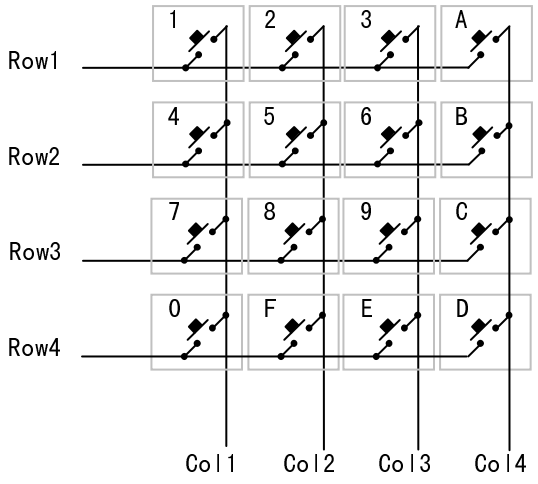
This experiment requires that you construct a circuit interfacing the MicroStamp11 module with a matrix keypad, and write assembly language code to detect when a button is pressed. If the pushed keypad button corresponds to keys 1 through 9 then an LED is blinked that many times.

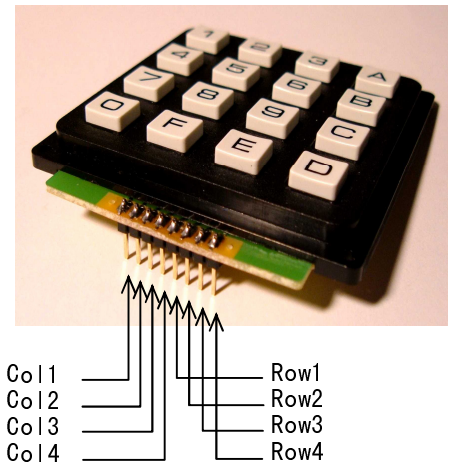
Matrix-style keypads use fewer wires than do individual switches, but this comes at the cost of added software complexity. Of course, the whole idea of using an embedded microcontroller instead of discrete logic is to shift the processing burden from hardware to software. A price is paid for complex hardware with every unit that is manufactured, whereas complex software costs the same as simple software to implement! Therefore it is important that you become familiar with microcontroller interfacing tasks such as this one.

Required Hardware

In addition to the MicroStamp11 module, for this experiment you will need the matrix keypad, 220Ω DIP resistors, 10KΩ SIP resistors, and one LED.

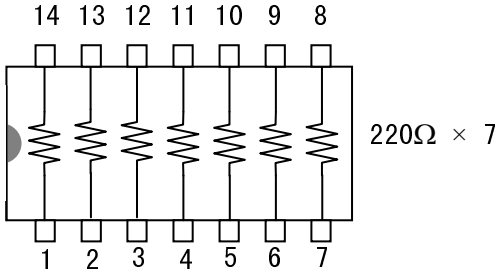
Here are the internal wiring and pin-out diagrams of the matrix keypad you will be using:



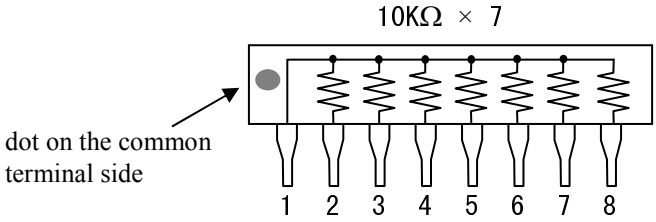


Each of the keypad pushbuttons activates a normally-open switch. Pressing a pushbutton closes a switch between the corresponding row and column wires. For example, pressing button 8 connects Row3 to Col2. The matrix arrangement reduces the number of wires required to interface with these switches from 17 (one per switch plus a common) to 8.

The Dual Inline Package (DIP) 220Ω resistor chip, which in your parts kit is blue in color, provides a convenient way of handling seven independent resistors:



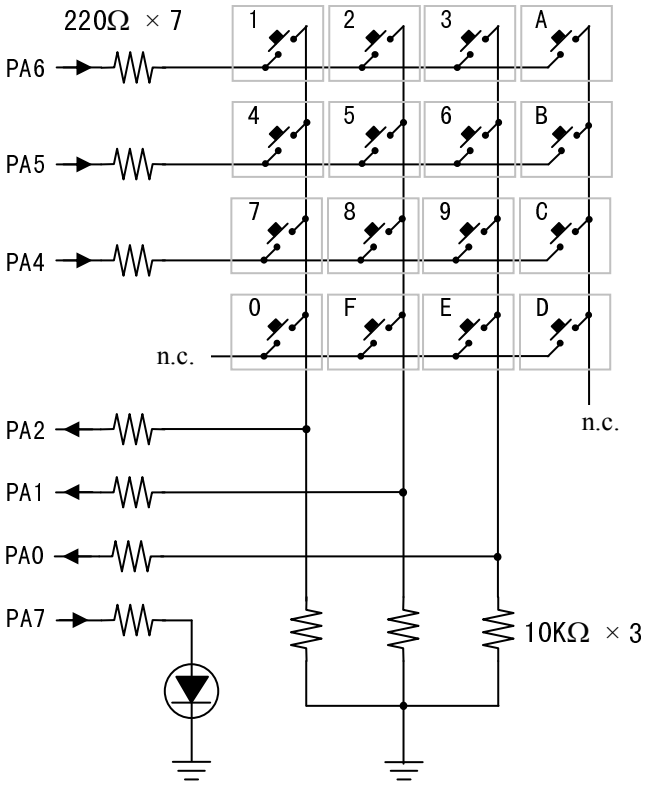
The Single Inline Package (SIP) 10KΩ resistor chip provides a convenient way of handling seven resistors that share a common terminal:



Wiring Diagram

Build the following circuit on a solderless breadboard.

(n.c. = not connected)



The top six 220Ω resistors serve the purpose to protect the microcontroller in case of wiring error, and the one tied to PA7 is a current limiting resistor for the LED. The 10KΩ resistors pull down to zero volts the matrix keypad column wires when no pushbutton is pressed.

To determine if pushbutton 8 is being pressed, for example, we first output {PA6, PA5, PA4} = {0, 0, 1}. Then read the value of PA1. Only if PA1 = 1 then 8 is being pressed. The code to do these checks is quite simple. Here is the code that implements this example:

```
; assume register X is initialized to Regbas

Poll_8:
     BCLR   PortA,X,%01100000              ; clear PA6,PA5
     BSET   PortA,X,%00010000              ; set PA4
     BRCLR  PortA,X,%00000010,Poll_9       ; if PA1=1 then
     JMP    Key8                           ;  jump to Key8

Poll9:  (etc.)
```

Then at label Key8, using timing loop techniques from Lab #2, you would flash the LED connected to PA7 eight times, wait for all keys to be released on the keypad, then go back to polling each key to see if one of them is pressed.

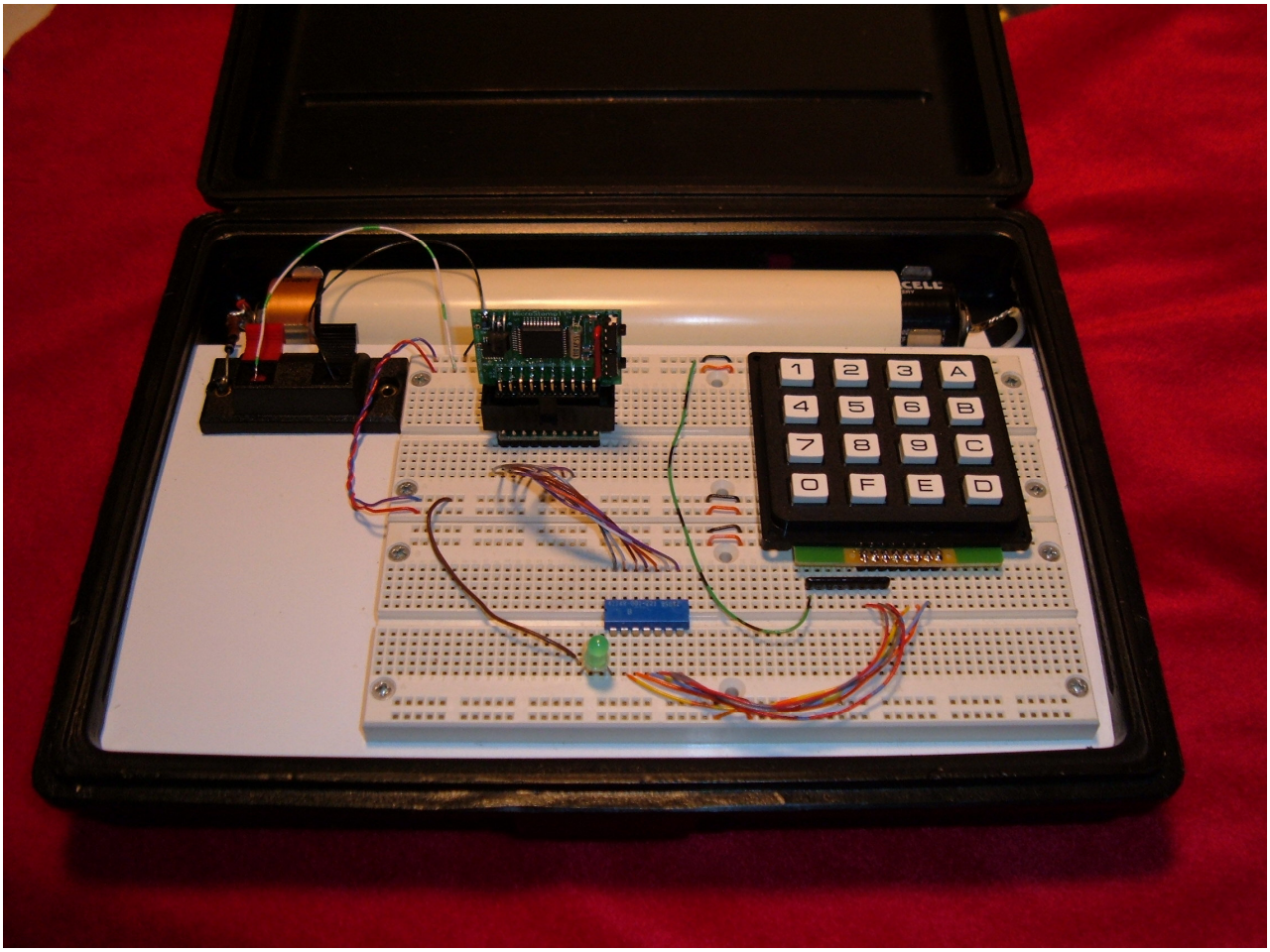
Here is some code that you may use to wait until the key being pressed has been released.

```
; Wait for keypad button to be released:

     BSET  PortA,X,%01110000          ; set PA6,PA5,PA4
     BRSET PortA,X,%00000100,*        ; wait for Col1 clear
     BRSET PortA,X,%00000010,*        ; wait for Col2 clear
     BRSET PortA,X,%00000001,*        ; wait for Col3 clear
```

The symbol ***** represents the address of that particular line of code, without having to define a label. For example, the following two lines of code result in identical machine code:

```
     JMP *

A0:  JMP A0
```

Write assembly language code to continuously poll the matrix keypad to detect if any key in the range 1 through 9 is pressed, and if one is pressed then to flash the LED at PA7 as many times as the key number. A good flashing period for the LED is 0.4 sec (0.2 sec on, 0.2 sec off). Demonstrate this circuit to your T.A.



Sample circuit layout for Laboratory Experiment #3.