Kit 71. PIC DUAL DICE

In an earlier Kit, Kit 69 we introduced the PIC microcontroller programmed as a single electronic dice. This Kit is an extension of the code to give two electronic dice. Two-dice games are very popular and we had a request to do it. All the commented software code for this Kit is available in a zip file **k71.zip** on our software download page at

http://kitsrus.com/soft.html

This code for the dice programs may be freely copied and modified.

The kit is constructed on a single-sided printed circuit board (PCB). Protel Autotrax and Schematic were used to produce the board.

ASSEMBLY INSTRUCTIONS

Check off the components in the kit against the Component listing. Solder in the resistors, the diode D1 and C2 first. Make sure to get the diode around the correct way; the black bar on the diode corresponds with the bar on the overlay. Next solder in the taller components; the IC socket, the 10uF electrolytic capacitor C1, voltage regulator, the LED display and the switches. Make sure the flat side of the hatkey switch corresponds to the flat shown on the overlay diagram of the switch. Insert the programmed PIC chip. Make sure the IC notch is towards the top of the PCB corresponding with the overlay.

SOFTWARE SPECIFICATION

We want to generate two random numbers between 1 and 6 everytime the Roll switch is pressed and to display the numbers on the LED displays for a few seconds. Then we want to be able to press Roll again and get another pair of numbers. If the Roll is not pressed for about a minute then we want the battery to turn off (that is, use the sleep mode of the PIC) to save power. Pressing the switch for about a second should wake the PIC from sleep mode and be ready to roll the 2 dice again. A slide switch is built into the circuit so that the unit can be turned completely off when not in use. This is because the voltage regulator does draw some small current when it is in sleep mode.

Hardware Setup. There are two I/O ports in the 16C54; one 8 bit & the other 4 bit. The 8 bit port, Port B, is connected to the 7 segment display. Three pins of the 4 bit Port A are used. The other pin is tied high. Pin 4 of the PIC is also tied high to give a power-on reset.

A 150R resistor acts as a current limiting resistor for each display. PIC ports have a maximum sourcing current of 40mA. (The maximum current for the LED display is 200mA so it is the PIC that R1 is protecting, not the display.)

The PIC uses its internal RC clock oscillator option (with the 10K resistor, R3, and the 330pF capacitor, C2) and runs at about 330kHz. The switch is connected to input RA0. Normally this line is pulled high via R4 to +5V, but when the switch is closed the input is pulled low. The software detects the falling edge as a dice roll.

Software Outline. Download the file **k71.zip** from our software download page on our website and print out the program listing. When power is first applied, the PIC does a power-on reset. It goes to Start. The first task is to initialize the various internal options such as the direction of the I/O ports and the status of any outputs (high or low.) The watchdog timer is switched on and the initial value of some RAM locations are set.

The dice now settles down in the main program loop, Main. It continuously loops about 4000 times a second to see if the switch has been pressed, flash the decimal point, generate the next random number and reset the watchdog timer. Then when a Roll (switch press) is detected the program jumps through to Main02. The two dice are rolled and displayed sequentially in Output. Note how the second random number is generated by the length of time the roll switch is depressed. Display outputs the random numbers to the 7 segment LED displays by multiplexing to them. At the end of Output the program returns to Main.

Various subroutines - Display, Delay, Roll, Flash, Random and two look-up tables for the display, Look_Up_Seg and Look_Up_Digit are listed in the final part of the program. Note the Rand01 code segment in Random and why it is there.

If there is no keypress for 50 flashes then the PIC goes into sleep mode at the end of the Flash subroutine. The oscillator is turned off and ports A & B are configured as inputs.

Now with the PIC in sleep mode the watchdog timer is not being reset anymore. It was not turned off when the PIC went to sleep. It now times out about every 1.3 seconds to check if the key is being pressed. If no keypress is found then it goes back to sleep. But if it is then the PIC wakes-up, resets the ports, starts to flash the decimal point and it is ready to roll again on the next keypress.

What to Do Now. You should have been able to identify in general terms the various parts of the program mentioned above in the program listing. Now pick out a part of the code and try to understand it in detail. Get the 33 PIC instructions and make sure you understand what is happening. This will immediately lead you into the inner workings of the chip. Remember you will not learn everything in one sitting. Try to find someone you can call on to ask questions & who will mentor you through this task.

WHAT TO DO IF IT DOES NOT WORK

Poor soldering is the most likely reason that the circuit does not work. Check all solder joints carefully under a good light. Next check that all components are in their

Kit 71. PIC DUAL DICE

correct position on the PCB especially the IC, electrolytic capacitor, the diode and the switch. Is the battery OK?

References
The Internet is where to find References. We have listed some on our links page at

http://kitsrus.com/back.html#pic

There is an active PIC List which you can join and ask questions.

You may email me at

peter@kitsrus.com

if you have any questions.

- - - - - - - - - - - - - - - - -

COMPONENTS	
150R resistor R1 R2	2
10K resistor R3 R4	2
10uF ecap C1	1
330pF ceramic C2	1
PIC 16C54C 04/P programmed IC	1
18 pin IC socket	1
Single-Digit LED display	2
78L05 voltage regulator	1
1N4148 diode D1	1
SPDT Switch	1
K71 PCB	1
Hatkey switch	1
9V battery snap	1