

Digital Signal Input and Output

Purpose

- To introduce the digital input and output functions of the Basic Stamp microcontroller
- To practice reading logic level signals from an input port
- To practice sending logic level signals to an output port

Components

<u>Qty.</u>	<u>Item</u>
1	OEM Basic Stamp II microcontroller and serial port cable
1	470 Ω DIP resistor pack
1	7-segment LED (common anode (CA))
2	tact switches
2	10 k resistors

Introduction

In this lab you will explore the input/output capability of the Basic Stamp microcontroller. Microcontrollers are inherently digital devices, meaning they operate with discrete values. The discrete values that all digital devices use are the binary values 0 and 1. The voltages 0 V and 5 V respectively usually represent these discrete values.

The Basic Stamp has 16 pins (P0 through P15) that can either serve as digital outputs, or as digital inputs. A digital output means that a program running on the Stamp can change the pin voltage to be either at common potential (0) or 5 V by writing a 0 or 1 to that pin. A digital input means that the world *outside* the microcontroller can change the voltage on the pin to either 0 V or 5 V, and the microcontroller can record the value as a 0 or 1 respectively.

7-segment LED display

You will use a 7-segment LED display as a digital output device and push-button switches as digital input devices. A 7-segment LED is nothing more than 7 LED's arranged in a pattern that can form a character when the appropriate segments are lit. These displays come in two basic types: common anode (CA) and common cathode (CC). CA types have all of the anodes of the 7 LED's connected together, and each of the 7 cathodes independent. Power is applied to the common anode, and a segment will be lit when its cathode is grounded. (Don't forget to use a current limiting resistor between the cathode and ground!) The reverse is true for the CC types. Figure 1 below shows a schematic diagram of a CA 7-segment LED. The letters as shown denote the particular segment. Two of the physical pins on the display are tied together (made 'common'). For the CA-type of display, the two pins connect to the common anode. For the CC-type of display, the two pins connect to the common cathode. The easiest way to figure out which pins correspond to which connections is to look at the data sheet for the device. Without a data sheet, you will need to "buzz" out the two common pins using the diode check function on a multimeter.

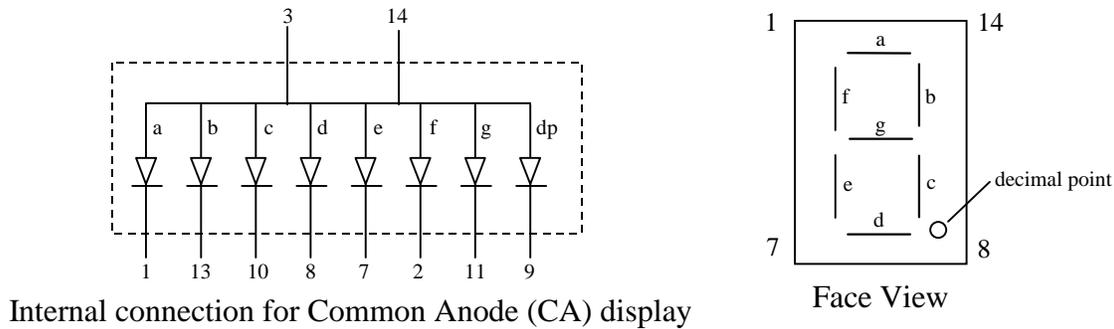


Figure 1 7-segment LED Display (Common Anode (CA) type). Two of the physical pins are tied together at the common anode of the display segments. The last LED is for the decimal point. Power is applied to the either of the common anode pins. A segment is lit when its cathode is then grounded through a current limiting resistor. Pin positions are numbered from 1 to 14, however some pins are not physically present on the actual device. See the data sheet for the particular device to find out which pin corresponds with which segment.

7447 BCD-to-7-segment (CA) display driver

The most common way 7-segment displays are implemented is with a BCD-to-7-segment decoder/driver chip. This chip takes a 4-bit binary number (like 0101, which corresponds to decimal 5) as an input, and when connected to a 7-segment display, it causes the proper LED segments to turn on and display the corresponding decimal number. The chip used with CA displays is the 7447. For CC displays it is the 7448. Figure 2 shows the pinout diagram for the 7447 and describes its operation. (The actual lettering on the chip may include other letters and numbers like, SN74LS47). By standard convention, pin 1 on any IC package is always the lower leftmost pin when the IC is oriented as shown with the U-shaped notch, or dot toward the left. Pin numbers procede to the right and loop around the right end of the chip as shown. Some IC's will only have the notch or dot; some have both

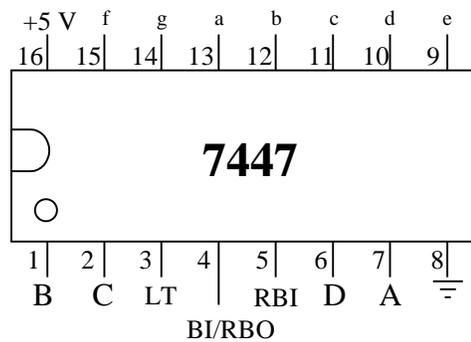


Figure 2 7447 BCD-to-7-segment decoder driver chip. This chip takes a 4-bit binary number applied to DCBA, where A is the least significant bit (LSB), and grounds the appropriate pin 9-15, so that when these pins are connected to a 7-segment LED display through current limiting resistor, the corresponding decimal number will appear on the display. The 7447 is used to drive common anode (CA) displays.

Switch input

You will use some switches to provide digital inputs. A switch is either on or off, hence it makes for a very simple digital sensor. You will use two momentary, normally open (NO)

single pole, single throw pushbutton switches called ‘tact’ switches. These are intended for soldering to PC boards, but by bending the leads properly, they can be inserted into a solderless breadboard. Looking on the underside of the switch, you can see a small dot molded into the plastic body. The two leads on the same side as the dot are tied together internally, and the two leads on the opposite side are tied together internally. When the button is pressed, electrical connection is made between the two sides. **BE CAREFUL** inserting the switch into the holes in the solderless breadboard. Make sure that each pair of legs that are tied together internally plug into the *same* row of 5 holes on the breadboard. The legs have already been twisted for you to make correct insertion into the breadboard easy. If you try to insert the switch rotated by 90°, you may damage the breadboard, so pay attention.

Procedure

Switch-Controlled Display Circuit

Figure 3 shows the circuit you will use in this lab.

1. Build the section of the circuit shown in the dashed rectangle (**A**) first. **Do not** connect the 7447 to the Stamp yet! (As we have emphasized in previous lab, you will save yourself lots of time, effort, and frustration by building and testing *pieces* of a complicated circuit rather than trying to wire up everything in one shot. So don’t rush. Build and test in small modules.) Also, liberties have been taken with the schematic to make it clear without having wires crisscrossing all over the place. Use the pinout diagrams for the 7447 and display to locate where the pins are actually located. Test circuit **A** by grounding the inputs DCBA. The decimal digit ‘0’ should appear on the 7-segment display.

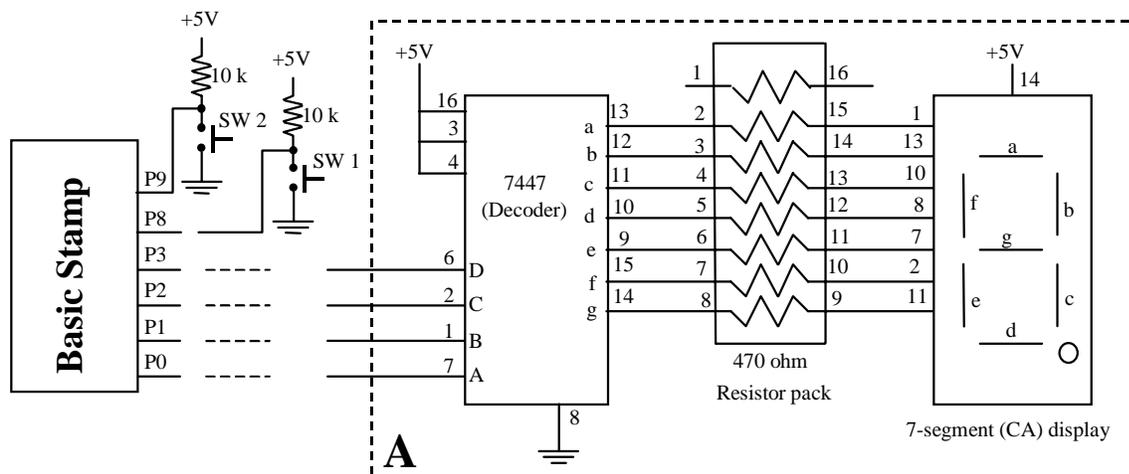


Figure 3 Switch-controlled display circuit.

2. After you have proven that the 7447 and 7-segment display have been wired correctly, connect power and common to the Stamp pins V_{in} and V_{ss} respectively. Don’t connect anything else to the Stamp yet! To make sure that the Stamp is not running some previous program, enter and run the following program:

```
'Stop everything
'Put your name here
'Put the date here
end
```

3. Now connect the DCBA inputs to the Stamp pins P3, P2, P1, and P0 respectively. Enter and run the following program:

```
'Display test
'Put your name here
'Put the date here
LOW 0
LOW 1
LOW 2
LOW 3
end
```

What does the 7-segment display show? The command LOW n does two things: 1) It makes the pin n an output, and 2) It takes the voltage down close to common potential (i.e., outputs a '0'). **What is the voltage at pins 9, 10, 11, 12, 13, and 15 of the 7447 chip? What is the voltage at pin 14 of the 7447 chip?** Save the program to your floppy disk.

Change the line "LOW 0" to HIGH 1. **Which pins of the 7447 do you expect to be at 5 V, and which are low?** Experiment with making pins P0-P3 HIGH or LOW until you are satisfied with your understanding of what is happening with the Stamp pins and your circuit.

4. The approach above is not very amenable to automating the output of a number to the display. Suppose you had a loop that counted between 0 and 9, and you wanted to output the count to the display. How would you do it? One way to do it is to store a binary number in a variable in the microcontroller's address space that corresponds to the I/O pins. The 0's in the binary number will set the corresponding pin low, and the 1's will set the corresponding pin high. Try the following program:

```
'Display loop test
'Put your name here
'Put the date here
counter var nib 'defines the variable counter as a 4 bit variable
OUTA=$0        'sets the lower 4 bits in the output register to b0000, i.e., makes sure P0-P3
                'will go low when DIR is set to output
DIRA=$F        'sets the lower 4 bits in the DIRS register to b1111, i.e., makes pins P0-P3
                'be outputs, so values in OUTA show up there. Should see '0' on display now

loop:
FOR counter=0 to 9
OUTA=counter
pause 1000
NEXT
goto loop
```

What happens to the display? Save the program to your floppy disk. **Modify the program to count down.**

So far we've been dealing with the digital output capabilities of the Stamp. Now let's bring in the digital input capabilities. Let's make the display circuit output "0" if neither button is pressed, "1" if SW1 is pressed, or "2" if SW2 is pressed.

5. Complete the circuit in Figure 3 by wiring in the two switches and their current limiting resistors. Try the following program:

```

'Switch Controlled Display Circuit
'Put your name here
'Put the date here
OUTA=$0      'sets the lower 4 bits in the output register to b0000, i.e., makes sure P0-P3
              'will go low when DIR is set to output
DIRC=$0      'sets the lower 4 bits of the upper byte in the DIRS register to b0000, i.e.,
              'makes P8-P11 inputs
DIRA=$F      'sets the lower 4 bits in the DIRS register to b1111, i.e., makes pins P0-P3
              'be outputs, so values in OUTA show up there. Should see '0' on display

loop:
IF IN8=1 AND IN9=1 THEN zero
IF NOT IN8=1 AND IN9=1 THEN one
OUTA=2
goto loop
one:
OUTA=1
goto loop
zero:
OUTA=0
goto loop

```

Explain how this program works following the label “loop:”

6. **Modify the program from Step 5 so that the display will output “3” if both switches are pressed (and held down)**
7. Optional exercise: Write a program that will cause the display to count up (continuously, with 1 second pauses between numbers) when SW1 is pressed and held down, and count down when SW2 is pressed and held down.

Food For Thought

How would you interface a dc motor and switch to the Stamp, so that you could turn the motor on when the switch is pressed? Draw a schematic for your interface and write a short program to implement your idea.