

Digital Signal Input and Output

Purpose

- To introduce the digital input and output functions of the OOPic 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	OOBOT 40-II microcontroller and serial port cable
1	470 Ω DIP resistor pack
1	7-segment LED (common anode (CA))
1	7447 BCD to 7-segment LED decoder IC
2	tact switches
2	10 k resistors

Introduction

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

The OOBOT 40-II microcontroller can service up to 29 digital inputs or outputs. A digital output means that a program running on the OOPic can change the pin voltage to be either at common potential (0) or at 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 light emitting diode (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.

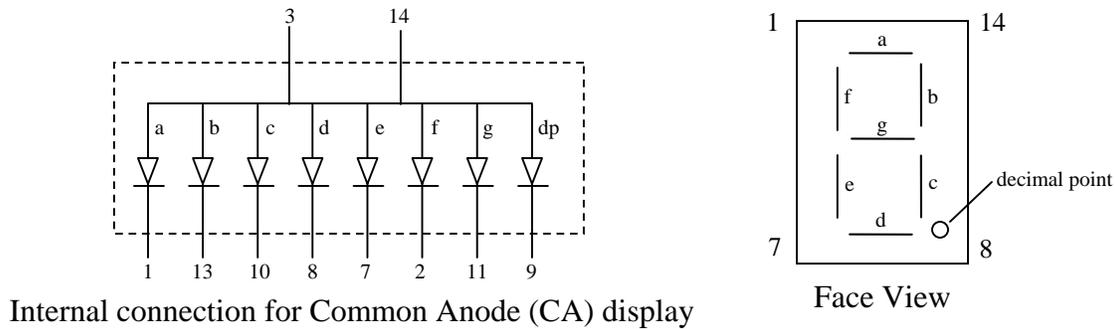


Figure 1 7-segment LED Display (Common Anode (CA) type). Two of the physical pins are tied together at the common anode of the display segments. The last LED is for the decimal point. Power is applied to the either of the common anode pins. A segment is lit when its cathode is then grounded through a 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 proceed 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

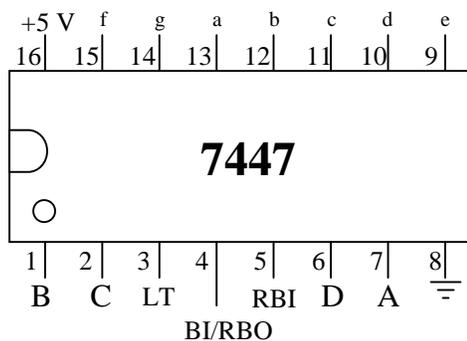


Figure 2 7447 BCD-to-7-segment decoder driver chip. This chip takes a 4-bit binary number applied to DCBA, where A is the least significant bit (LSB), and grounds the appropriate pin 9-15, so that when these pins are connected to a 7-segment LED display through current limiting resistor, the corresponding 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 OOPic yet! (As we have emphasized in previous labs, 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 shown in Appendix A for the 7447 and the LED display to see where the pins are actually located. Test the circuit in **A** by grounding the inputs DCBA. The decimal digit '0' should appear on the 7-segment display.

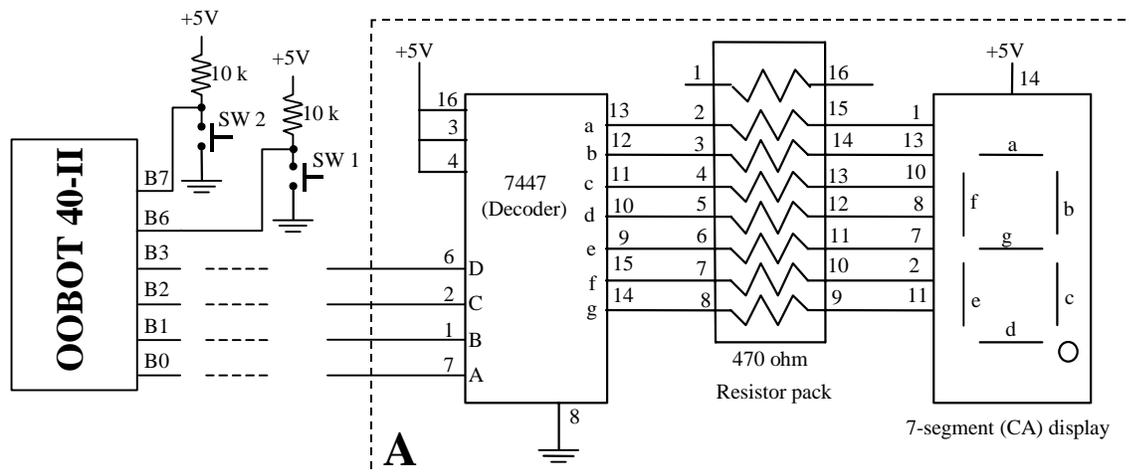


Figure 3 Switch-controlled display circuit. Two tact switches provide digital inputs to the OOPic, which in turn drives (by four digital outputs) a 7447 seven-segment decoder IC. Note the 10 k Ω resistors connected to the switches. These are called 'pull up' resistors, because they pull the voltage of pins B7 and B6 up to 5 volts when the switch is not pressed. They also limit the current to ground when the switches are pressed.

2. After you have proven that the 7447 and 7-segment display have been wired correctly, connect power and the serial cable to the OOBOT board. Don't connect anything else to the OOPic yet! Note that when you power up the OOPic it will run the program you

downloaded last, so it is a good idea to enter some benign program (e.g., just comments), such as:

```
// Display Test Program 1
// Put your name here
// Put the date here
```

before you start connecting outputs that might be sending signals that you don't expect.

3. Now connect the DCBA inputs to the OOPic pins B3, B2, B1, and B0 respectively. Enter and run the following program:

```
// Display Test Program 1
// Put your name here
// Put the date here

oDio4 Display = new oDio4;      // Declare 4-bit digital I/O object
sub void main(void)
{
  // Initialize oDio4 object

  Display.IOGroup=1;          // Use pins in IO Group 1 (B0 – B7)
  Display.Nibble=cvLow;      // Use only the lower 4 in the IO Group (B0 – B3)
  Display.Direction=cvOutput; // Make pins digital outputs

  Display=0;                  // Make pins B0 – B3 low
}
```

What does the 7-segment display show? 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.

Modify the program to write a '1' to Display instead of '0'. **Which pins of the 7447 do you expect to be at 5 V, and which are low?** Experiment by writing numbers between 0 and 15 until you are satisfied with your understanding of what is happening with the OOPic pins and your circuit.

4. Modify the program so that the display will count from 0 to 9 continually with a delay of 0.75 seconds in between numbers.

So far we've been dealing with the digital output capabilities of the OOPic. 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:

```
// Display Test Program 2
// Put your name here
// Put the date here

oDio4 Display = new oDio4;      // Declare a 4-bit digital I/O object for 7447 interface
oDio4 Switches = new oDio4;    // Declare a 4-bit digital I/O object for interface to the switches
sub void main(void)
{
  // Initialize Display object

  Display.IOGroup=1;          // Use pins in IO Group 1 (B0 – B7)
```

```

Display.Nibble=cvLow;           // but only use the lower 4 in the IO Group (B0 – B3)
Display.Direction=cvOutput;     // Make pins digital outputs

// Initialize Switches object

Switches.IOGroup=1;            // Use pins in IO Group 1 (B0 – B7)
Switches.Nibble=cvHigh;       // but only use the upper 4 pins in the IO Group (B4 – B7)
Switches.Direction=cvInput;    // Make pins digital inputs

while(1)
{
    // Test for switch presses and write to display
    switch (Switches.Value)
    {
        case 8:                 // Switch 1 is pressed (pin B6 is low, B7 is high)
            Display=1;          // Write a '1' to the 7447
            break;
        case 4:                 // Switch 2 is pressed (pin B6 is high, B7 is low)
            Display=2;          // Write a '2' to the 7447
            break;
        default:                // Default condition
            Display=15;         // Blank the display
    }
}
}

```

Explain how this program works following the while(1) statement.

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 OOPic, 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.

Appendix A – Pinout for 7447

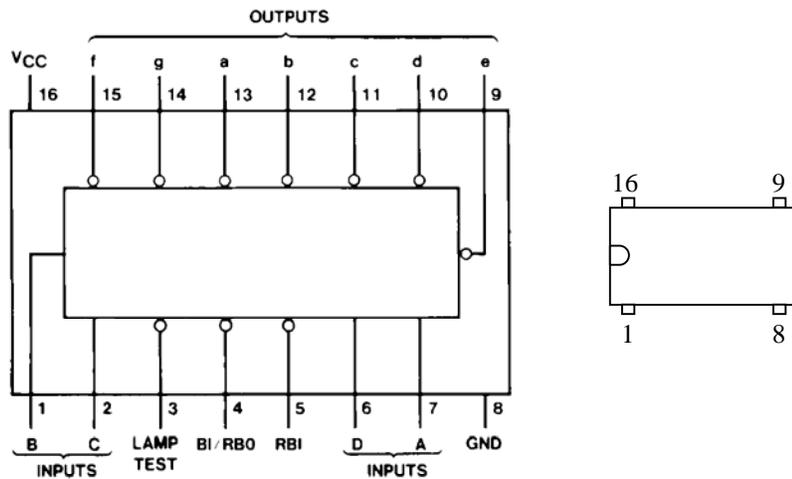


Figure A1 Pin diagram for 7447. Pin 1 can be located by finding the U-shaped depression or dot on the package. (Connection diagram from Fairchild Semiconductor, <http://www.fairchildsemi.com/ds/DM/DM7447A.pdf> [visited on 13SEP03].)

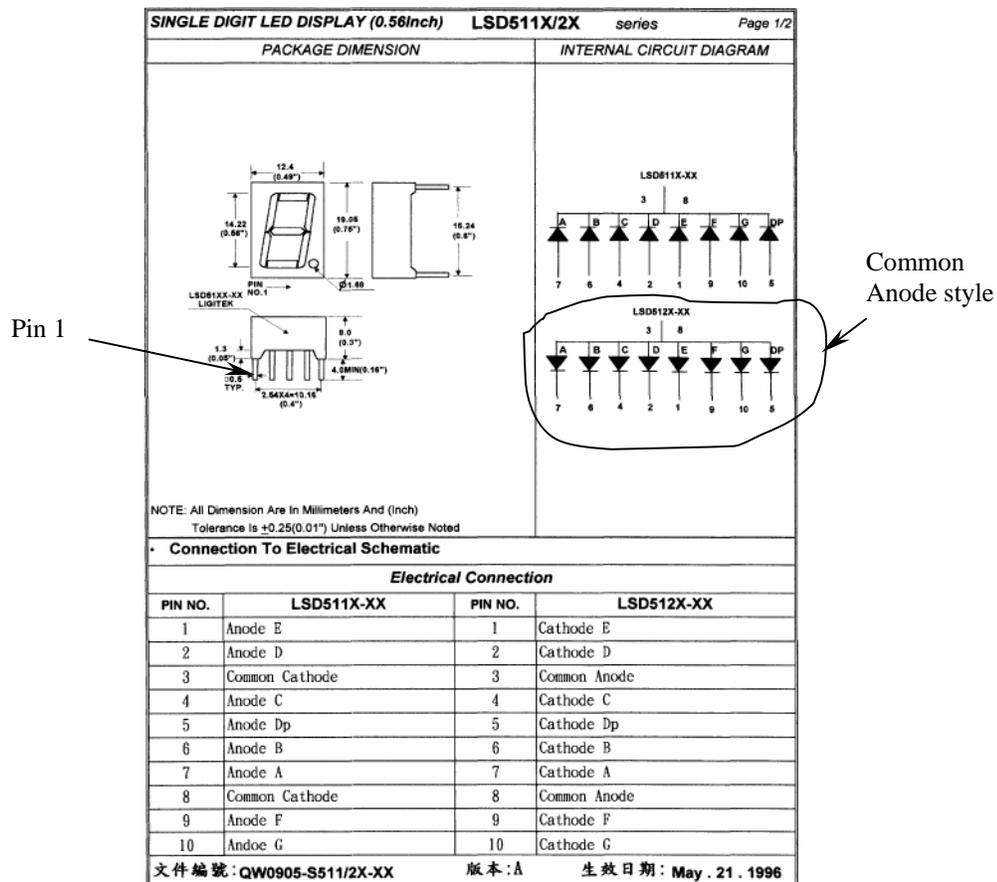


Figure A2 Pin diagram for Ligitek LSD512X series 7-segment LED (Jameco part no. 104213, <http://www.jameco.com/jameco/Products/ProdDS/104213.pdf>, [visited on 13SEP03].)

Function Table

Decimal or Function	Inputs						BI/RBO (Note 1)	Outputs							Note
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g	
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H	(Note 2)
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H	
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L	
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L	
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L	
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L	
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L	
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H	
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L	
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L	
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L	
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L	
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L	
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L	
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L	
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H	
BI	X	X	X	X	X	X	L	H	H	H	H	H	H	H	(Note 3)
RBI	H	L	L	L	L	L	L	H	H	H	H	H	H	H	(Note 4)
LT	L	X	X	X	X	X	H	L	L	L	L	L	L	L	(Note 5)

H = HIGH level, L = LOW level, X = Don't Care

Note 1: BI/RBO is a wire-AND logic serving as blanking input (BI) and/or ripple-blanking output (RBO).

Note 2: The blanking input (BI) must be OPEN or held at a HIGH logic level when output functions 0 through 15 are desired. The ripple-blanking input (RBI) must be OPEN or HIGH if blanking of a decimal zero is not desired.

Note 3: When a LOW logic level is applied directly to the blanking input (BI), all segment outputs are HIGH regardless of the level of any other input.

Note 4: When ripple-blanking input (RBI) and inputs A, B, C, and D are at a LOW level with the lamp test input HIGH, all segment outputs go H and the ripple-blanking output (RBO) goes to a LOW level (response condition).

Note 5: When the blanking input/ripple-blanking output (BI/RBO) is OPEN or held HIGH and a LOW is applied to the lamp-test input, all segment outputs are L.

Figure A3 Logic diagram for 7447 Decoder/Driver IC.
 (<http://www.fairchildsemi.com/ds/DM/DM7447A.pdf>, [visited on 13SEP03].)