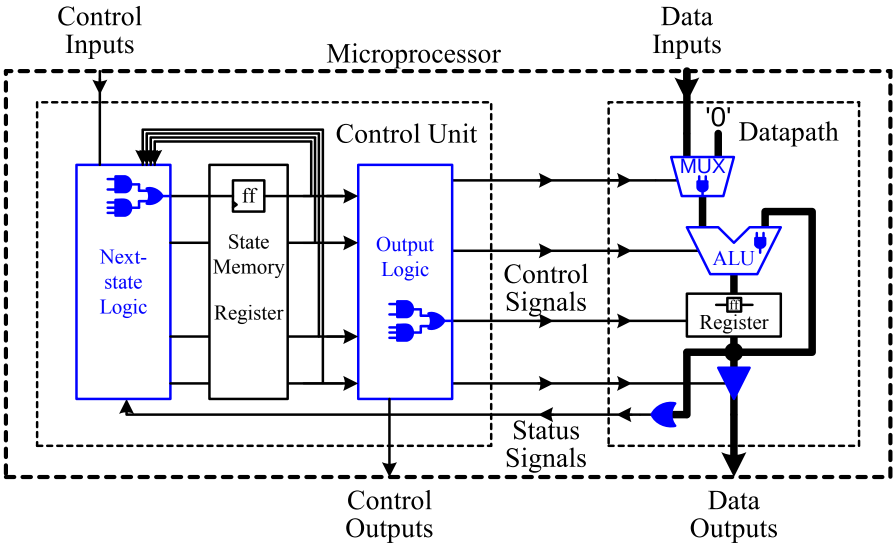
Basic Microprocessor Interfacing Trainer

Lab Manual



Enoch Hwang, Ph.D.

Table of Contents

1 Microprocessor Interfacing Trainer

We know that microprocessors are at the heart of all electronic devices. We know that they are responsible for manipulating the data and controlling the entire operation of the device. In the Microprocessor Trainer (DL-030), you have learned how to design microprocessors. However, having just a microprocessor by itself is totally useless if it cannot interact with the world that we live in. In order for microprocessors to do their intended job, they must be able to get inputs from the real world, and after processing the data, be able to provide the outputs back to the real world.

In this Microprocessor Interfacing Trainer series, we will look at the various methods in which we can provide inputs to the microprocessors and outputs from the microprocessors. Some common output devices include light emitting diodes (LED), seven-segment LED displays, relays, liquid crystal displays (LCD) and VGA monitors. Some common input devices include switches, push buttons, keypads, and various kinds of environmental sensors. In this Basic Trainer, you will learn how to interface simple input/output devices. In the Advance Microprocessor Interfacing Trainer, you will learn how to interface more complex input/output devices. In order for you to implement some of the input/output circuits discussed in this trainer, you will need in addition to the components that are included in this trainer, the DL-030 Digital Logic Trainer.

2 Light-Emitting Diodes (LED)

2.1 Introduction

A light-emitting diode (LED) is a small light source made of semiconductor material. LEDs are often small in area, and are typically mounted in a plastic case with a round lens on top as shown in Figure 1. Early LEDs only emitted low-intensity red light, but today, LEDs are available in many colors and with very high brightness.

Figure 1. Light-Emitting Diode.

Compare to traditional incandescent light source, LEDs use lower energy, have a longer lifetime, are smaller in size, and are able to turn on and off at a much faster rate. Because of these advantages, they are used in many lighting applications. Furthermore, infrared LEDs can also be used in communications such as remote control units for televisions.

2.2 Physical Connections

LEDs have two connections, an anode (+) and a cathode (–) as shown in Figure 2. The longer connection post is the anode and the shorter connection post is the cathode. On some LEDs, the plastic by the cathode connection post has a flat spot. The schematic diagram for a LED is shown in Figure 2 (c).

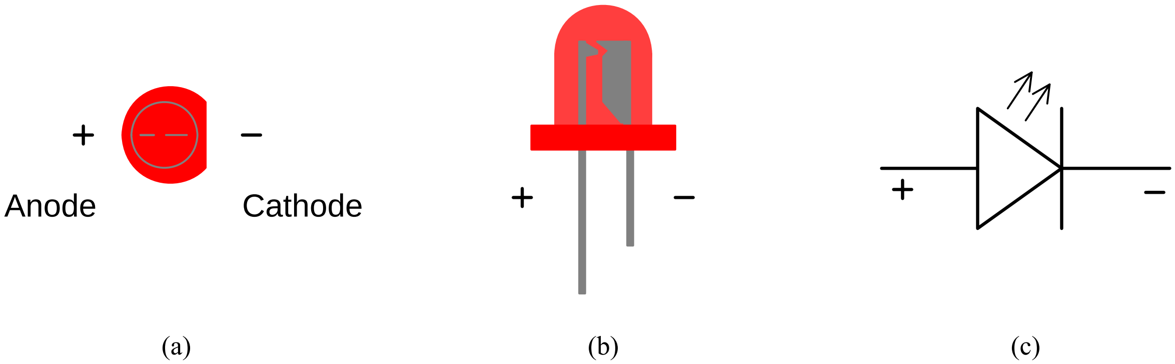


Figure 2. LED: (a) top view; (b) side view; (c) schematic diagram.

A typical LED can handle a maximum forward current of about 30mA. To prevent it from being destroyed, the LED must be connected in series with a resistor before connecting it to a 5V DC power source. A typical safe resistance value to use is 330Ω. To be more precise with the current and the resistor value, you would need to look up the specification for that particular LED that you are using to see what the maximum current that the LED can handle. Figure 3 shows the schematic of how a LED is to be connected to a 5V DC power source. The negative end of the LED is connected to the negative end of the battery. The positive end of the LED is connected, via the 330Ω resistor, to the positive end of the battery. It will also work if the resistor is connected between the negative end of the LED and the negative end of the battery.

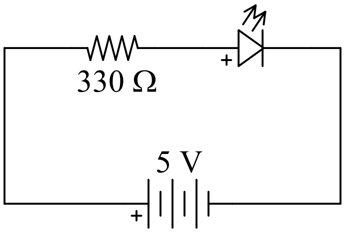


Figure 3. Connection of an LED to a 5 V battery.

When using a LED in a digital circuit, you would want your circuit to act as a switch to turn on and off the LED. The output of a logic gate can provide enough current to drive a LED. Figure 4 shows how an LED is connected to a two-input AND gate. The negative end of the LED is connected to ground. The positive end is connected via a resistor to the output of the AND gate. In this circuit, the AND gate is acting as a simple switch. One input of the AND gate is connected directly to VCC. The second input to the AND gate is used to turn on and off the LED. When this input has a logic 1 the LED will turn on, and when this input has a logic 0 it will turn off.

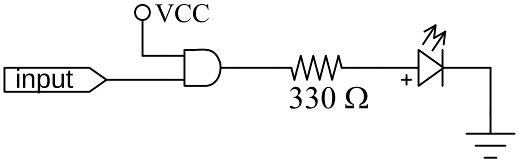


Figure 4. Controlling an LED with an AND gate.

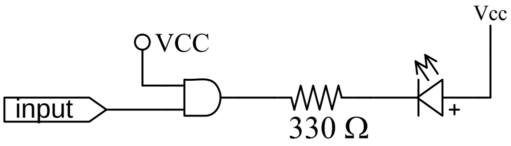
2.3 Experiments

1. Connect the LED circuit as shown in Figure 3 to confirm that it works.

2. Connect the LED circuit as shown in Figure 3 but with the resistor connected between the negative end of the LED and the negative end of the battery. Confirm that this also works.

3. Connect the circuit as shown in Figure 4 to confirm that it works. To turn on the LED with a logic 1, connect the input pin to VCC. Connect the input pin to ground to turn it off.

4. In the following circuit, what logic value should the input be in order to turn on the LED? Connect the circuit to confirm your answer.

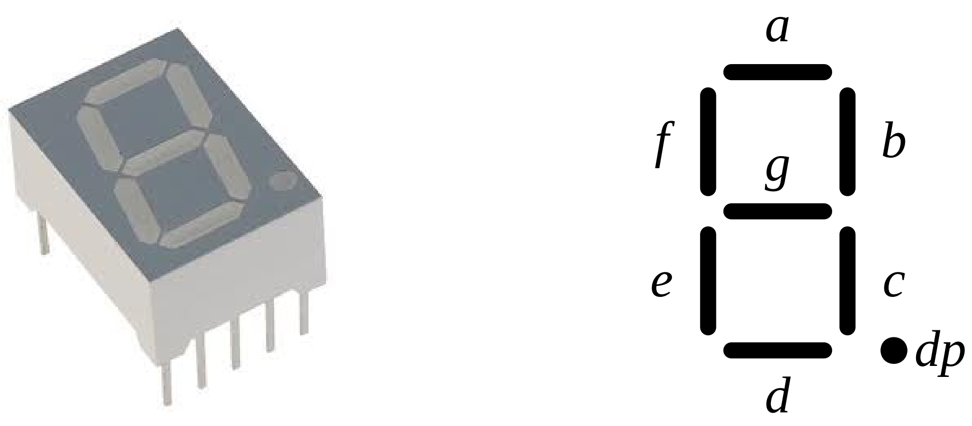


5. Connect a LED on the DL-030 trainer's breadboard. Design and implement a controller circuit on the DL-030 to turn on and off the LED at a rate of approximately 1 second interval.

3 Seven-Segment Displays

3.1 Introduction

A seven-segment display is a device that is composed of seven elongated individual LEDs arranged like the figure 8 with optionally another round LED for the decimal point as shown in Figure 5 (a). Each of the LED (also referred to as a segment) is usually given a letter name from *a* to *g* for the seven segments as shown in Figure 5 (b).



(a) (b)

Figure 5. Seven-segment display: (a) picture; (b) letter names assigned for the seven segments and the decimal point.

Seven-segment displays are usually used to show decimal numbers. For example, to show the number 0, segments *a*, *b*, *c*, *d*, *e* and *f* are turned on while segment *g* is turned off. And to show the number 1, segments *b* and *c* are turned on while all of the others are turned off. Figure 6 shows how the ten decimal digits are shown on a seven-segment display.

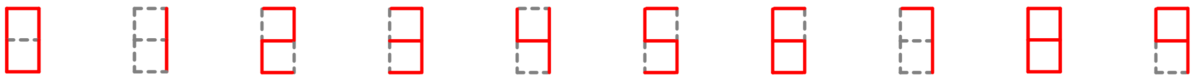
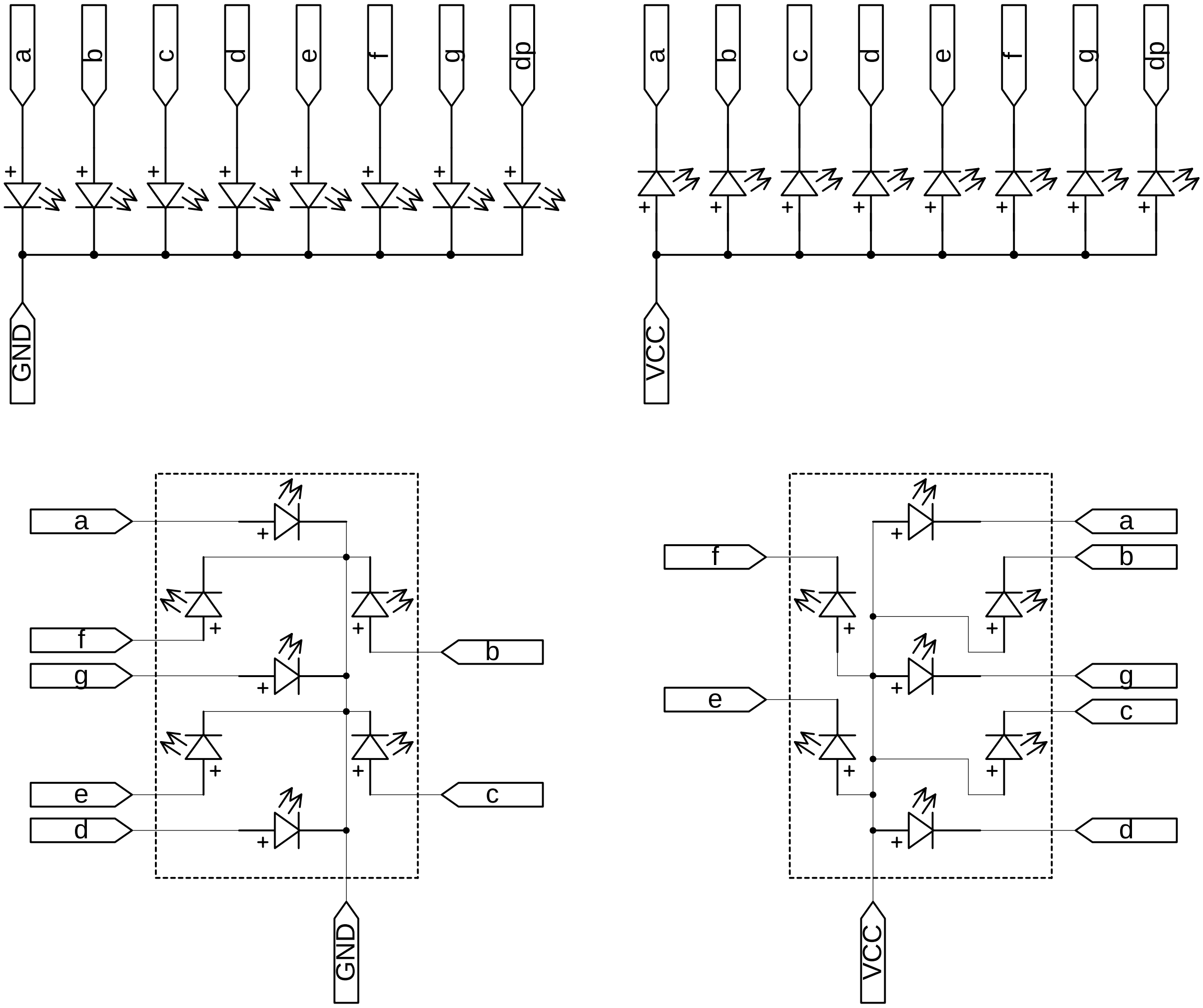


Figure 6. Seven-segment display showing the ten decimal digits.

Other variations of the seven-segment display are the 14-segment and 16-segment displays for showing alphanumeric characters. There are also dot matrix displays with a grid of 7 × 5 round LEDs.

3.2 Physical Connections

As discussed in Section 2.2, each LED has two connections. At first thought, one would think that a seven-segment display that has a decimal point will, therefore, have 16 connections for the eight LEDs. However, it turns out that only nine connections are needed for the eight LEDs because one connection from each LED can be connected together in common. To do this, either the anode or the cathode of each of the eight LEDs is connected together in common. The schematic drawing of how the eight LEDs are connected to form the seven-segment display with the decimal point is shown in Figure 7. Figure 7 (a) shows a common cathode connection and Figure 7 (b) shows a common anode connection.



(a)

(b)

Figure 7. Internal connections of the seven-segment display: (a) common cathode; (b) common anode.

Just like for a single LED, a resistor connected in series is needed to prevent from too much current passing through each LED. Usually only one resistor is needed for each display if it is connected to the common end of the display.

3.3 Experiments

1. Connect a seven-segment display to turn on all of the segments. Remember to connect a resistor in series with the common connection.

2. Is there a difference in the brightness of the display if one resistor is connected at the common end versus using eight resistors each connected to each of the segment end?

3. Connect a seven-segment display to turn on, one at a time, each of the ten decimal digits.

4. Connect a seven-segment display on the DL-030 trainer's breadboard. Design and implement a controller circuit on the DL-030 to display the count from zero to nine on a seven-segment display.

4 Multiplexing Seven-Segment Displays

4.1 Introduction

Typically two or more seven-segment displays are combined together in a package so that a larger decimal number can be displayed as shown in Figure 8.

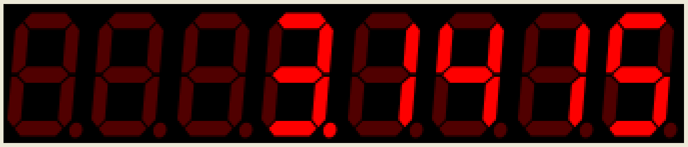


Figure 8. Picture of eight seven-segment displays in one package showing the number 3.1415.

4.2 Physical Connections

Individually the eight seven-segment display package shown in Figure 8 would require 72 (8 × 9) connections. In order to reduce the number of connections, the eight seven-segment displays are connected in such a way that fewer connections are needed. Internally, all of the same segments for each display are connected together in common. In other words, segment *a* for all of the displays are connected together, segment *b* for all for the displays are connected together, etc. As a result, no matter how many digits are in a package, there will only be eight connections for the seven segments and the decimal point. And then for each display there will either be a common cathode or anode connection. Figure 9 shows the internal connections of a three-digit seven-segment display with a common cathode.

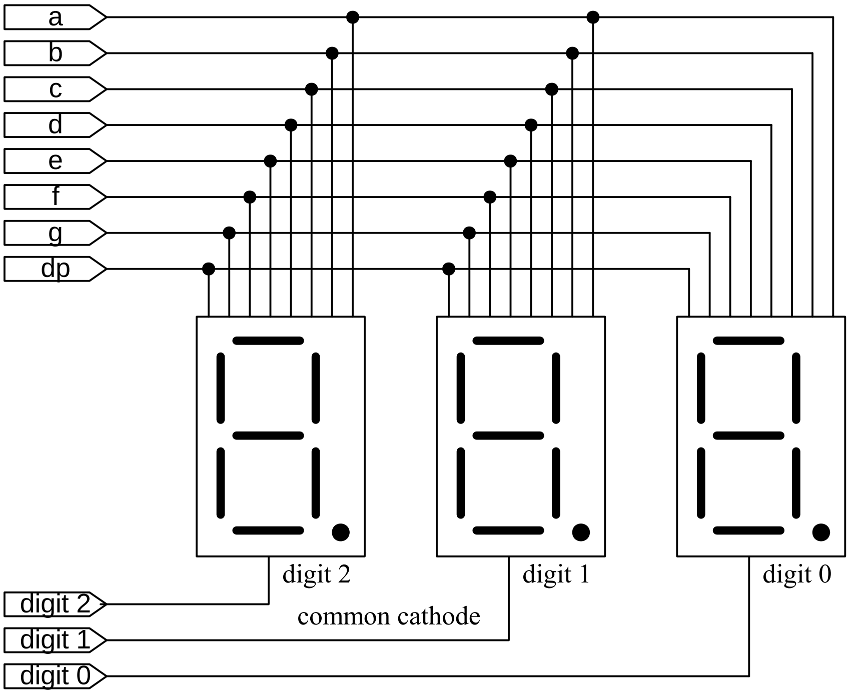


Figure 9. Internal connections of a three-digit seven-segment display with common cathode.

For a common cathode display as shown in Figure 9, a particular LED is turned on by connecting the segment connection to a 1 and the digit connection to a 0. Because all of the same segments from the digits are connected together, therefore, if you connect, say, segment *a* to a 1 and all of the three digit connections to a 0, then segment *a* for all three digits will be turned on. The problem now is how to turn on a particular segment on one digit and turn off the same segment on another digit. For example, to display the two digits 10, we need to turn off segment *a* for digit 1 and turn on segment *a* for digit 0. But in order to turn on both digits, the two digit connections must be connected to 0.

The solution to this problem is known as time multiplexing. Time multiplexing turns on only one digit at a time for only a short period of time. This will continuously cycle through turning on one digit at a time for all of the digits in the display. Although the digits are being turned on one at a time, but because they are cycling through so fast, we will get the impression that all of the digits are turned at the same time. The net result, of course, is that each digit will be dimmer then if it is continuously on.

4.3 Experiments

1. Connect a two-digit seven-segment display to turn on all of the segments for both digits. Remember to connect a resistor in series with the common connection.

2. Is it possible to manually connect a two-digit seven-segment display so that you can see the two decimal digits 10 on the display? Why yes or why not?

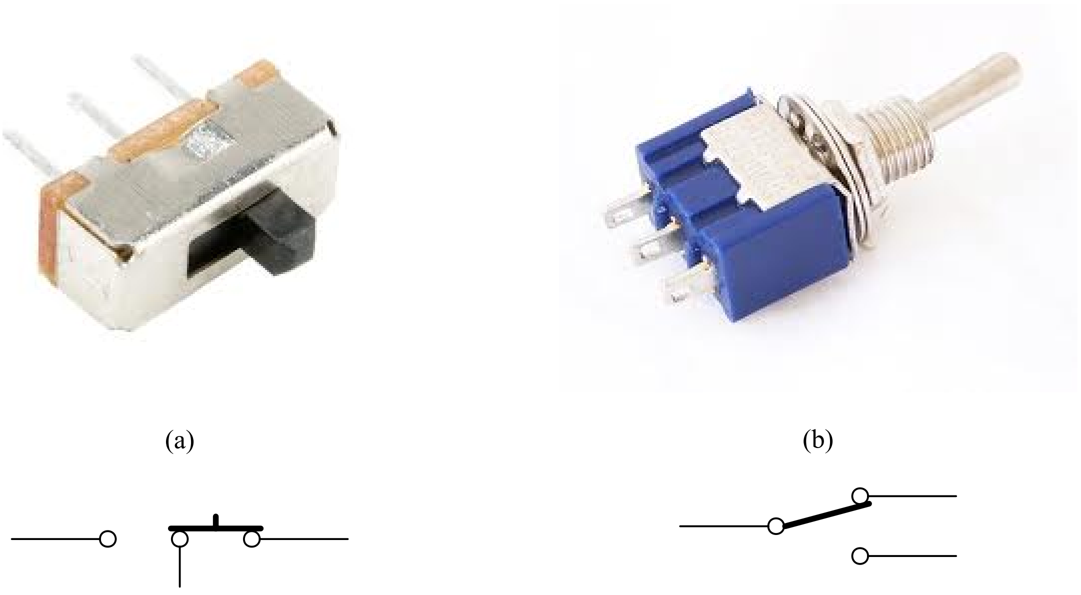
3. Connect a two-digit seven-segment display package on the DL-030 trainer’s breadboard. Design and implement a controller circuit that will display the two decimal digits 10.

4. Connect a two-digit seven-segment display package on the DL-030 trainer’s breadboard. Design and implement a controller circuit that will display the count from zero to ninety nine.

5 Switches

5.1 Introduction

A switch is an electromechanical device that can be switched between two positions. An electrical connection is made between a common point and either one of the two positions that the switch is in. These switches are known as binary switch because of the two different states that they can be in. Switches come in many different styles, two of which (the slider switch and the toggle switch) are shown in Figure 10 (a) and (b). Their corresponding schematic diagrams are shown in Figure 10 (c) and (d).


(a) (b)

(c) (d)

Figure 10. Two different styles of switches: (a) slider switch; (b) toggle switch; (c) schematic diagram for the slider switch; (d) schematic diagram for the toggle switch.

For the slider switch, the mechanical part can be slide back and forth making connection between the common point with either the left connection point or the right connection point. The toggle switch operates exactly the same except that the switch handle can be flipped up and down.

5.2 Physical Connections

Regardless of whether it is a slider or toggle switch, functionally they are the same. There are three connections with one being the common. The switch connects between the common connection with either one of the other two connections. A typical connection for a slider switch is shown in Figure 11 (a). The left connection point is connected to VCC to provide a logic 1 signal and the right connection point is connected to GND to provide a logic 0 signal. The center common connection point is connected to your data input. When the switch is in the left position, a logic 1 is seen at the common point, and when the switch is in the right position, a logic 0 is seen at the common point. In Figure 11 (b), the LED will be turned on when the switch is in the up position and off when the switch is in the down position.

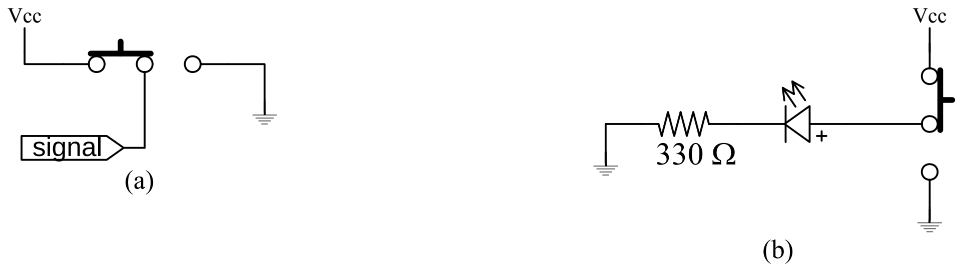


Figure 11. (a) A typical connection for a slider switch; (b) connecting the switch to a LED.

5.3 Experiments

1. Connect the switch and LED circuit as shown in Figure 11 (b) to confirm its operation.
2. Modify the circuit in Figure 11 (b) so that the LED is turned on when the switch is in the down position.

6 Push Buttons

6.1 Introduction

A push button is also an electromechanical device like a switch except that it always automatically goes back to either the on or off position. Because of this phenomenon, push buttons are classified as either normally opened or normally closed. For a normally-opened push button, the electrical connection is normally in the off position (no contact) when the button is not pressed, and when the button is pressed the electrical connection is made and current can flow through. When the button is released, the switch will return back to the off position. For a normally-closed push button, it is just the opposite. The electrical connection is normally in the on position (contact made) when the button is not pressed, and when the button is pressed the electrical connection is broken and current cannot flow through. When the button is released, the switch will return back to the on position. Figure 12 (a) shows a picture of a typical push button. Figure 12 (b) shows the schematic diagram for a normally-opened push button and Figure 12 (c) shows the schematic diagram for a normally-closed push button.

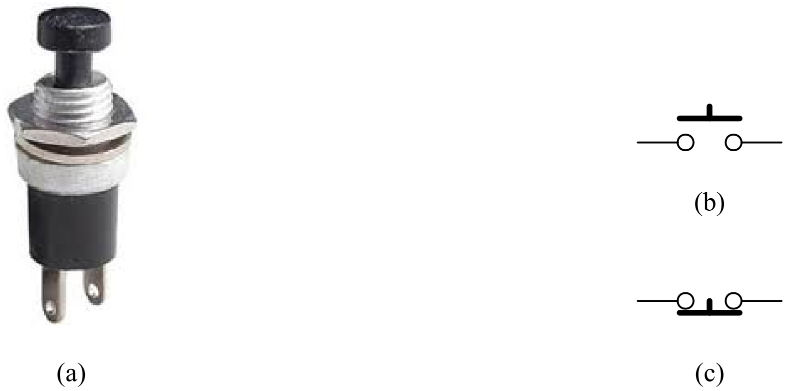


Figure 12. Push button: (a) picture; (b) schematic diagram of a normally-opened push button; (c) schematic diagram of a normally-closed push button.

6.2 Physical Connections

Another difference between a push button and a switch is that a push button has only two connection points whereas a switch has three connection points. Having only two connection points, it might not be obvious as to how a push button can be connected to produce both a logic 1 and a logic 0 signals. For example, if we connect a normally-opened push button circuit as shown in Figure 13 (a) then the signal point will have a logic 1 when the button is pressed; and if we connect the circuit shown in Figure 13 (b) then the signal point will have a logic 0 when the button is pressed. However, in both cases, the question we want to ask is what is the logic value when the button is not pressed? With the two circuits shown in Figure 13 (a) and (b), we cannot assume that the logic value is the opposite value when the button is not pressed. In other words, the logic value is undefined when the button is not pressed.

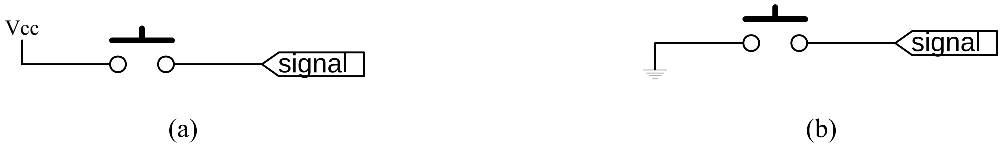


Figure 13. (a) Incorrect circuit that only results in a logic 1 signal when the button is pressed; (b) incorrect circuit that only results in a logic 0 signal when the button is pressed.

It is extremely important that for all inputs to any digital circuit, the logic value cannot be undefined; it must be either a logic 1 or a logic 0. Both of the circuits in Figure 13 will result in an undefined logic value when the button is not pressed. In order to resolve this problem with the push button, we need to connect the circuit shown in Figure 14. In Figure 14 (a) there is a 22K Ω resistor connected between the signal point and ground. This resistor is often referred to as a pull-down resistor. What this resistor will do is that when the button is not pressed, it will provide an electrical path from ground to the signal point, thus giving a logic 0 at the signal point. When the button is pressed, it will give a logic 1 at the signal point because of the direct connection to Vcc, but the resistor will prevent a short circuit between Vcc and Gnd. Figure 14 (b) shows basically the same idea but with a pull-up resistor to Vcc instead, and the switch is connected directly to Gnd.

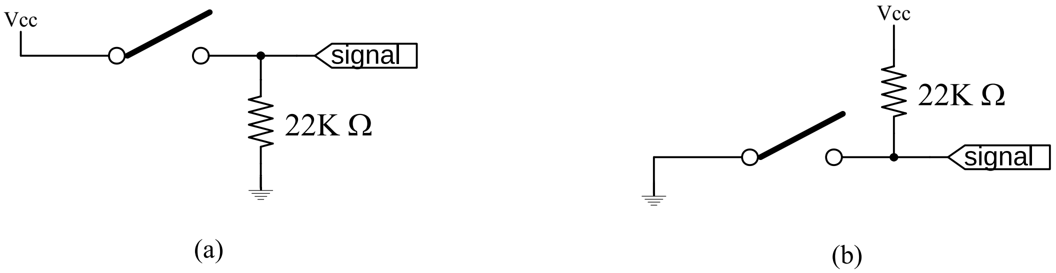


Figure 14. (a) Push button circuit with a pull-down resistor; (b) Push button circuit with a pull-up resistor.

6.3 Experiments

1. Connect the normally-opened push button circuit shown in Figure 14 (a). Connect the signal point to an LED so that the LED will be turned on when the push button is depressed. Remember to use a resistor in series with the LED.

2. Connect a normally-opened push button circuit to an LED so that the LED is turned off when the button is depressed.

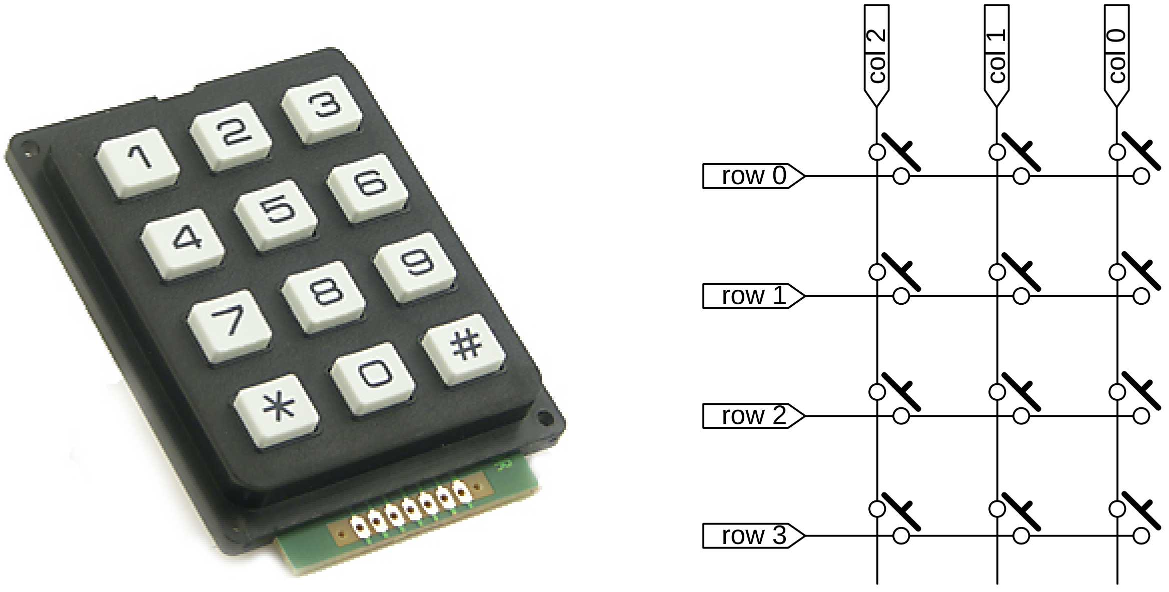
3. Connect a normally-closed push button circuit to an LED so that the LED is turned on when the button is depressed.

4. Connect a normally-closed push button circuit to an LED so that the LED is turned off when the button is depressed.

7 Keypads

7.1 Introduction

Keypads are just like push buttons but having several of them packaged together into one unit. Figure 15 (a) shows a picture of a standard 12 key keypad arranged in a 3×4 grid. Typically, normally-opened push buttons are used for the keys. Figure 15 (a) shows a picture of a typical 3×4 keypad.



(a)

(b)

Figure 15. Keypad: (a) picture; (b) schematic diagram.

7.2 Physical Connections

Normally for twelve discreet push buttons, there will be 24 connections since each push button requires two connections. In order to reduce the number of connections, the keys in a keypad are connected such that those in the same row will have one connection point connected in common and those in the same column will have the other connection point connected in common. The internal connections of the 3×4 keypad is shown in Figure 15 (b). With this connection configuration, there will only be seven connections for the twelve keys: three connections for the three columns and four connections for the four rows.

Using normally-opened keys, the intersections for each column and row are normally disconnected. When a particular key is pressed, it will connect between the column and the row that that key is located. For example, when key 1 is pressed, it will connect row 0 with column 2; when key 8 is pressed, it will connect row 3 with column 1.

We will now look at how we can connect a keypad so that we can distinguish which key has been pressed. Again we need to keep in mind that regardless of whether a key is pressed or not, we cannot have undefined logic values. As a first attempt, we might be tempted to do something similar to the circuit in Figure 14 (a) and end up with the circuit shown in Figure 16 (a). We first noticed that at the three column connection points, no undefined logic value is possible. For example, at column 2, when none of the keys in that column is pressed, then the signal is a logic 0 because of the pull-down resistor. However, when any one of the keys in that column is pressed, then column 2 will have a logic 1 because of the connection to Vcc. The problem, however, is that we would not know which of the four keys in that column has been pressed.

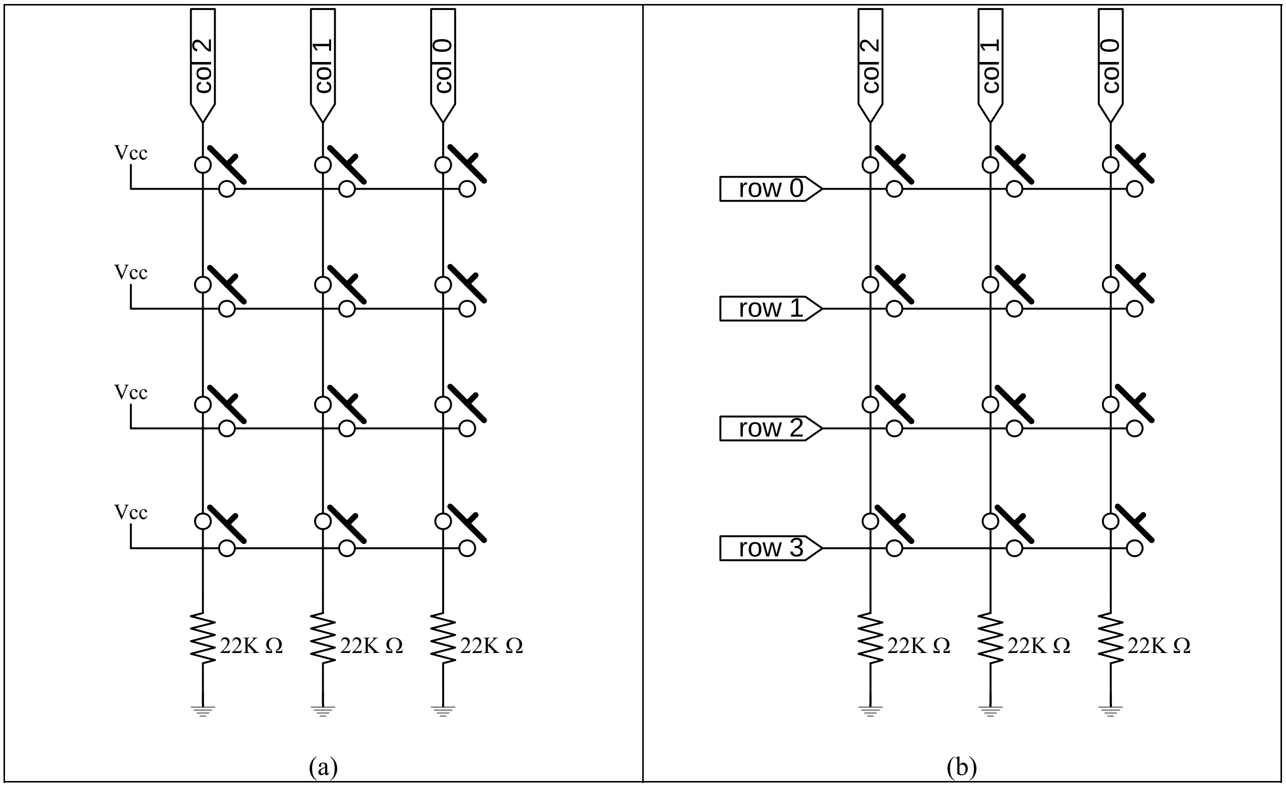


Figure 16. Circuit for interfacing keypad to microprocessor.: (a) incorrect; (b) correct

In order to distinguish which one of the four keys in the same column has been pressed, we cannot have all of the rows connected to Vcc. Just like in multiplexing the seven-segment digits, we need to selectively set one row at a time to the logic 1 value (i.e., Vcc) while the rest of the rows are set to a logic 0. This will cycle at a relatively fast speed with respect to the time it takes to press and release a key. To achieve this, we need to modify the circuit to be like that shown in Figure 16 (b). Replacing the Vcc connections for the four rows with just an input connection, these four inputs can now be set either to a logic 0 or a logic 1. So, for example, when key 0 is pressed, a logic 1 will be seen at the column 2 connection point if the row 0 connection is a logic 1. At this point you might wonder how is this different with the original circuit where all of the rows are connected to a logic 1 because for both cases you see a logic 1 at the column connection point when any one of the four keys is pressed? Well, the important difference is that if we set row 0 to a logic 0, column 2 will have a logic 0 regardless of whether key 0 is pressed or not. In other words, since we know which row we are sending out a logic 1, and if we get a logic 1 at column 2, then we know which intersection point in that column got connected by a key press.

For this connection scheme to work, we need to be able to send either a logic 1 or a logic 0 to each of the rows at a fast speed. This is achievable only by using a microprocessor to control it. The microprocessor will repeatedly cycle through each of the four rows by sending a logic 1 value to it while sending a logic 0 to the rest of the rows. After sending a logic 1 for each row, it will read the three column signals to see if any one of them has a logic 1 value. If there is a logic 1 value for a particular column, then the key at that intersection of the grid has been pressed.

7.3 Experiments

1. Connect a 3×4 keypad to the DL-030 trainer's breadboard. Design and implement a controller circuit that will be able to tell which key on the keypad has been pressed.

2. Combine the keypad controller from question 1 with the seven-segment display controller from Section 3 so that when a key is pressed the correct digit is shown on the seven-segment display.

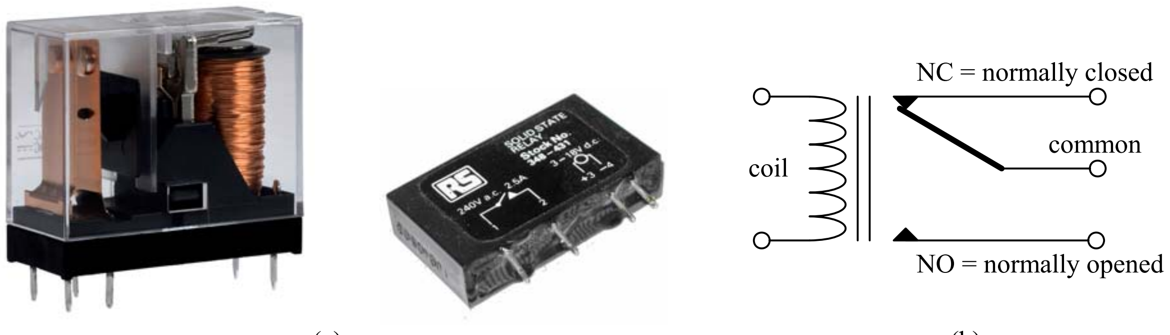
8 Relays

8.1 Introduction

The LED displays that we have looked at so far require low voltage and low current to turn them on. These low voltage/current devices can be controlled directly from the outputs of digital logic gates. However, there are many other devices that we want to control that require much higher voltage and/or current. Relays are electromechanical devices that allow a microprocessor circuit to control devices that require higher voltage or current. Figure 17 (a) shows two pictures of a relay. The first picture has a clear plastic housing allowing you to see the inside with the switch on the left and the magnetic coil on the right. The second relay in the picture is a solid state relay that replaces the magnetic coil and mechanical switch with solid state components.

A relay is composed of two parts. The first part is just like a switch with terminals that can handle high voltage and current. However, instead of being able to manually move the switch, the switch is operated by an electromagnetic coil. This electromagnetic coil forms the second part of the relay. The coil can be energized by low voltage and current, and thus, can be turned on and off by the outputs of a microprocessor circuit. The low voltage/current electromagnet controls the switch, which in turn controls the operation of high voltage/current devices. The schematic diagram for the relay is shown in Figure 17 (b).

The newer opto-isolated solid state relay functions identical to the mechanical coil relay. However, instead of using an electromagnetic coil to turn on and off a mechanical switch, an optical circuit is used so that the control portion of the relay is electrically isolated from the load portion, and instead of using a mechanical switch, a solid-state switch is used. The control voltage energizes an LED which illuminates and switches on a photo-sensitive diode; the diode current then turns on either a thyristor, silicon controlled rectifier, or MOSFET to switch a high voltage/current load.



(a) (b)

Figure 17. Relay: (a) picture; (b) schematic diagram.

8.2 Physical Connections

To interface a relay to a microprocessor, we want the relay coil to be controlled by the microprocessor. There are relays with coils that can be energized with very low current so that a microprocessor's output can be connect directly to the coil. In most cases, however, we might need a transistor to amplify the control signal and have the transistor's output drive the coil. Figure 18 shows a relay circuit using a general purpose NPN transistor (Q1) to drive the coil. The collector (C) output of the transistor is connected directly to one end of the coil. The emitter (E) is connected to ground, and the base (B) of the transistor is the control signal from the microprocessor. A typical general purpose NPN transistor is the 2N2222 or 3904.

A serious problem when working with an electromagnetic coil is that when the current to the coil is suddenly turned off, a very high voltage spike occurs. The magnetic field around the coil collapses, and it tries to keep the current flowing in the same direction as it was when it was powered, thus causing a very high reverse voltage. This high voltage is prevented when a reverse diode (D1) is connected in parallel with the coil, thus providing a path for the reverse current. Be careful that the cathode end of the diode (the end with the bar) must be connected to the power (5V) end of the coil, otherwise you will have an even more serious problem by creating a short circuit. A typical diode is the 1N4148 or 1N914.

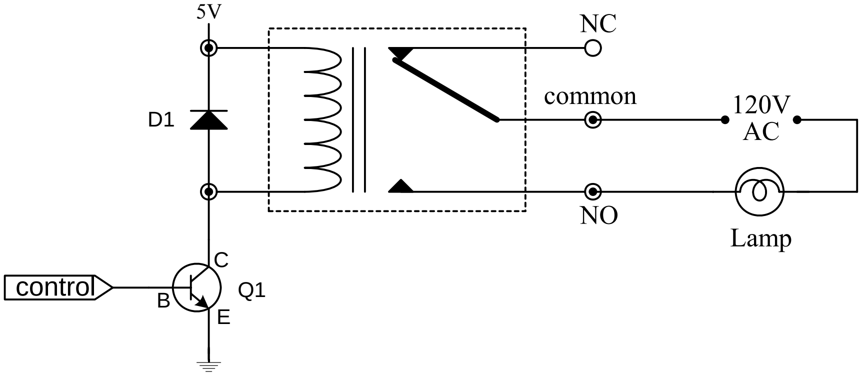


Figure 18. Circuit for controlling a relay by a microprocessor to turn on and off a 120V AC lamp.

The relay can now turn on most any high voltage load. Just make sure that your load is within the maximum ratings for the relay. The circuit in Figure 18 shows a 120V AC lamp connected to the relay switch. With this circuit, a microprocessor can turn on a 120V AC lamp by sending a logic 1 signal to the control pin.

8.3 Experiments

1. Connect the relay circuit shown in Figure 18 on the DL-030 trainer's breadboard. **Be extremely careful when connecting the 120V AC**. Test the circuit by manually connecting the control input pin to +5V DC to turn on the lamp.

2. Design and implement a controller circuit on the DL-030 that interfaces to the control pin of the relay circuit shown in Figure 18. The controller circuit will flash the lamp on and off in approximately two second intervals.

9 Light Sensors

9.1 Introduction

A photoresistor, also known as a light dependent resistor (LDR) is a resistor whose resistance decreases with increasing light intensity. A photoresistor is made of a high resistance semiconductor. If light falling on the device is of high enough frequency, photons absorbed by the semiconductor give bound electrons enough energy to jump into the conduction band. The resulting free electrons conduct electricity, thereby lowering resistance. Figure 19 shows a picture of a photoresistor.

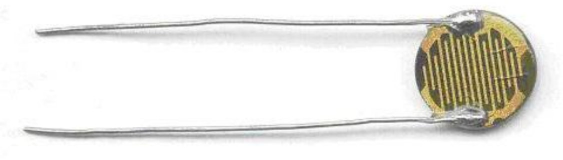


Figure 19. Picture of a photoresistor.

9.2 Physical Connections

A photoresistor can be used to detect the presence of light or no light since it has a very high resistance when there is no light and very low resistance when there is light. With this change in resistance, a photoresistor can be thought of as an on-off switch where having light will turn on the switch and no light will turn off the switch. The circuit shown in Figure 20 for interfacing a photoresistor is similar to the circuit for a push-button switch.

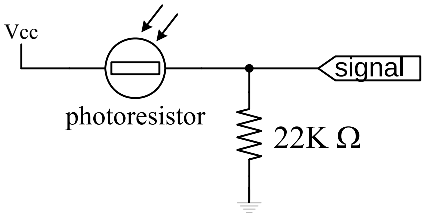


Figure 20. Circuit for interfacing a photoresistor.

When there is no light and the photoresistor has a very high resistance, the pull-down resistor will provide a logic 0 to the microprocessor, and when there is light, current will flow from Vcc through the low resistance photoresistor to the signal pin giving it a logic 1 value.

9.3 Experiments

1. Connect the photoresistor circuit shown in Figure 20 on the DL-030 trainer’s breadboard. Design and implement a controller circuit on the DL-030 that will sense the presence of light. The controller will turn on an LED when there is light and turn off the LED when there is no light.

10 Pressure Sensors

10.1 Introduction

A force-sensing resistor is a resistor whose resistance changes when a force or pressure is applied. A force-sensing resistor consists of a conductive polymer, which changes resistance in a predictable manner following the application of force to its surface. Applying a force to the surface of a sensing film causes particles to touch the conducting electrodes, thus changing the resistance of the film.

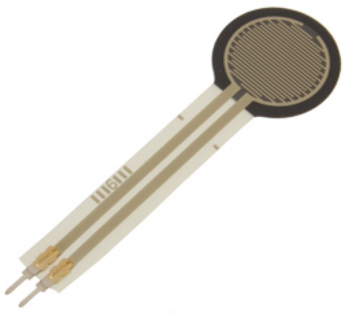


Figure 21. Picture of a force-sensing resistor.

10.2 Physical Connections

A force-sensing resistor can be used to detect the presence of a pressure since it has a very high resistance when there is no pressure and very low resistance when there is pressure. With this change in resistance, a force-sensing resistor can be thought of as an on-off switch where having a pressure will turn on the switch and no pressure will turn off the switch. The circuit shown in Figure 22 for interfacing a force-sensing resistor is similar to the circuit for a push-button switch.

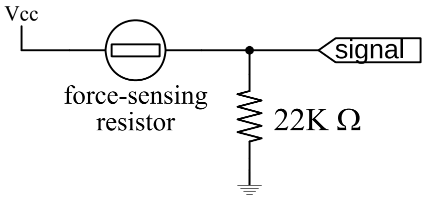


Figure 22. Circuit for interfacing a force-sensing resistor.

When there is no pressure and the force-sensing resistor has a very high resistance, the pull-down resistor will provide a logic 0 to the microprocessor, and when there is pressure, current will flow from Vcc through the low resistance force-sensing resistor to the signal pin giving it a logic 1 value.

10.3 Experiments

1. Connect the force-sensing resistor circuit shown in Figure 22 on the DL-030 trainer’s breadboard. Design and implement a controller circuit on the DL-030 that will sense the presence of a pressure. The controller will turn on an LED when there is pressure and turn off the LED when there is no pressure.

11 Temperature Sensors

11.1 Introduction

A thermistor is a type of resistor whose resistance varies with temperature. Thermistors typically operate within a temperature range of −90 °C to 130 °C with high precision. There are two types of thermistor: a positive temperature coefficient thermistor where the resistance increases with increasing temperature, and a negative temperature coefficient thermistor where the resistance decreases with increasing temperature. In situations where you need to measure temperatures outside of this range or you need more precision there are other temperature sensors such as the thermocouples or integrated circuit sensors like the LM35.

Figure 23. Picture of a thermistor.

11.2 Physical Connections

A thermistor can be used to detect a change in temperature. For a negative temperature coefficient thermistor, it has a very high resistance when it is cold and very low resistance when it is hot. With this change in resistance, a thermistor can be thought of as an on-off switch where a hot temperature will turn on the switch and a cold temperature will turn off the switch. The circuit shown in Figure 24 for interfacing a thermistor is similar to the circuit for a push-button switch.

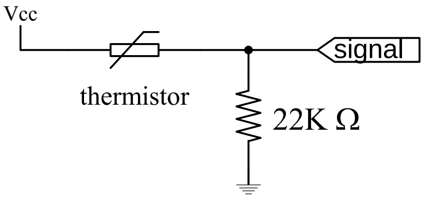


Figure 24. Circuit for interfacing a thermistor.

When the temperature is cold and the thermistor has a very high resistance, the pull-down resistor will provide a logic 0 to the microprocessor, and when the temperature is hot, current will flow from Vcc through the low resistance thermistor to the signal pin giving it a logic 1 value.

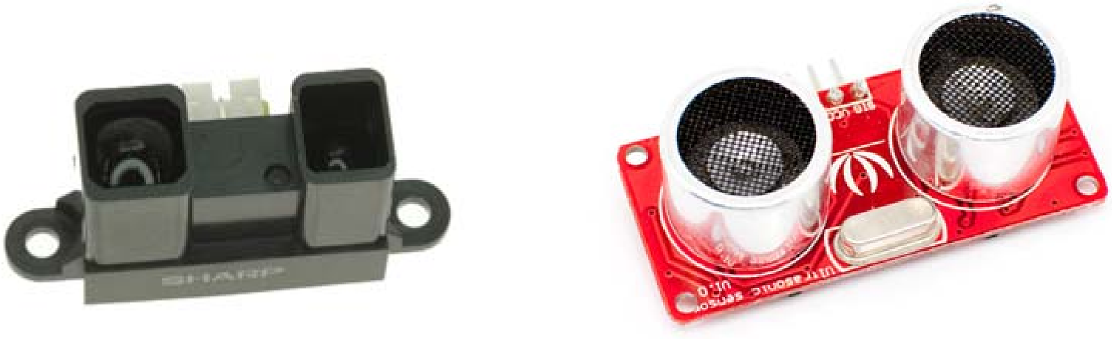
11.3 Experiments

1. Connect the thermistor circuit shown in Figure 24 on the DL-030 trainer’s breadboard. Design and implement a controller circuit on the DL-030 that will sense a hot or cold temperature. The controller will turn on an LED when it is hot and turn off the LED when it is cold.

12 Object Presence/Distance Sensors

12.1 Introduction

An object presence sensor, also known as a proximity sensor, is able to detect the presence of nearby objects without any physical contact. In operation, a proximity sensor continuously emits an infrared red (IR) beam. Upon hitting a nearby object, the infrared red beam is bounced back. The proximity sensor looks for changes in this return signal to determine whether there is an object or not in its line of sight. A typical infrared proximity sensor can sense objects from about an inch to a few feet. The Sharp IR distance sensor (GP2Y0A02YK) can report a distance range of 8 to 60 inches. Figure 25 (a) shows a picture of an infrared red proximity sensor.



(a)

(b)

Figure 25. (a) Picture of an infrared red proximity sensor; (b) picture of an ultrasonic distance sensor.

An ultrasonic sensor can be used to detect objects that are further away. The principle of operation is basically the same as the infrared sensor except that an ultrasonic beam is used instead of an infrared red beam. Typically, an ultrasonic sensor has a detecting range from an inch to 12 feet. Figure 25 (b) shows a picture of an ultrasonic sensor.

Some proximity sensors will report not just whether there is an object in its line of sight, but also the distance from the sensor to the object. The distance is reported as an analog voltage which varies linearly with the distance. Unlike the three passive environmental sensors from the last three sections, the distance sensor has some active electronics built into the device. Specifically, the circuitry needs to continuously send out an IR (or ultrasonic) beam and then calculates the elapsed time until the return signal is detected. This elapsed time is converted to a linear voltage available on its data output pin.

12.2 Physical Connections

The proximity or distance sensor usually has a 3-wire interface: power, ground and the output signal. The output signal will have a variable voltage that correlates to the distance to the object. For simplicity, we will ignore the variable voltage but just look at whether the voltage is above or below a certain threshold. This way, we will not be able to tell the distance between the sensor and the object but just whether there is an object or not. The interfacing circuit is shown in Figure 26. The value of the resistor will determine the threshold point. The higher the resistance the lower the threshold point.???

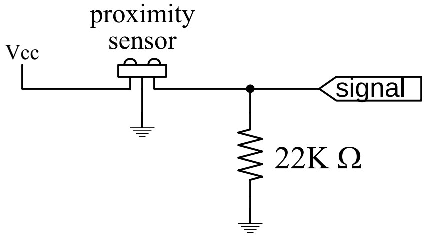


Figure 26. Circuit for interfacing a proximity sensor.

12.3 Experiments

1. Connect the proximity sensor circuit shown in Figure 26 on the DL-030 trainer's breadboard. Design and implement a controller circuit on the DL-030 that will sense the presence of an object or not. The controller will turn on an LED when there is an object in front of the sensor and turn off the LED when there is no object.

2. Repeat experiment 1 using different values for the resistor. Plot a graph between the resistor value and the threshold distance in which the LED will turn on.

13 Appendix

13.1 Parts List for the Experiments

Red LED

Single-digit seven-segment LED display

Two-digit seven-segment LED display

Two position slider switch

Normally-opened push button

3×4 keypad

Relay

9V motor

Photoresistor

Force-sensing resistor

Thermistor

Sharp IR distance sensor GP2Y0A02YK

Resistors: 100K Ω, 56K Ω, 22K Ω × 4, 10K Ω, 330 Ω × 8,

Transistor: NPN 2N2222

Diode: 1N4148