ZBasic

Application Note

AN-213 External Device Interfacing

Introduction

In most microcontroller projects, you will want to connect external devices to the ZX processor. Examples of such devices include LEDs, switches, relays, solenoids, piezo buzzers, etc. This application note discusses the basic concepts for connecting such devices. Once you understand the fundamental issues, you'll be able to apply the concepts to other devices that aren't specifically described.

I/O Pin Fundamentals

Each of the ZX I/O pins may be configured to be either an input or an output. When configured as an input a built-in pullup resistor (20K to 50K ohms) may optionally be enabled. Enabling the built-in pullup may eliminate the need for an external pullup resistor in some cases but if you need a "stronger" pullup, you'll have to add an external resistor.

When configured to be an output, the output level may be set to either logic zero or logic one. When set to logic zero, a pin may sink (take in) a maximum of 20mA.   When set to logic one, a pin may source (send out) a maximum of 20mA. The source and sink currents can be pushed as high as 40mA but it is advised to limit it to 20mA or less.

In addition to the per-pin limits described above, there are limits on the total source current and total sink current that can be handled by the device at any instant in time. These limits need to be considered whenever devices are connected to the processor. For the ZX-24 and ZX-44, the total source current for all of the pins of each of the ports A, B, C and D is limited to 100mA as is the total sink current. For the ZX-40, Port A has the same limits but ports B, C and D have an aggregate limit of 100mA combined source current and 100 mA combined sink current. The source current limits are restricted even further if the on-board regulator of the ZX-24 is being used. In that case, the total source current of all ports is limited to no more than 40mA.

If you need to connect a device to a ZX that exceeds the source or sink capability of an I/O pin it cannot be connected directly. It can, however, be connected using a transistor, relay, opto-isolator or other switching device as an intermediary. The same is true for devices that are not compatible with the 0 to 5 volt levels on the ZX I/O pins.

Connecting an LED

An LED, or Light Emitting Diode, is a device that converts electrical current into light. The light emitted by an LED might be in the visible spectrum or it might be either infrared or ultraviolet. In either case, the operation is the same – a current must flow through the LED in order to cause it to emit light. Generally, the larger the current, the brighter will be the emitted light.

The symbol that represents an LED on a circuit schematic is shown below. The symbol is the same as for an ordinary diode with the addition of the arrows that suggest light emission.

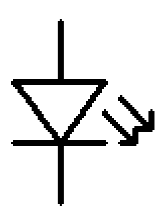

Figure 1 – Schematic Symbol for an LED

The two terminals of an LED are called the anode and the cathode. In Figure 1, the anode is the upper lead and the cathode is the lower one. Like other diodes, an LED can be thought of as allowing current to flow in one direction only – from the anode to the cathode. The direction of current flow can be remembered most easily by associating the arrowhead with the direction of allowed current flow. The bar at the tip of the arrowhead suggests that current is blocked from flowing in the opposite direction.

The LED offers little resistance to current flow in the "forward" direction (anode to cathode) but offers high resistance to current flow in the "reverse" direction (cathode to anode). Because of the low forward resistance, a resistor must be employed to limit the amount of forward current because too much current may destroy the LED. The schematic diagram in Figure 2 shows a simple circuit for illuminating an LED.

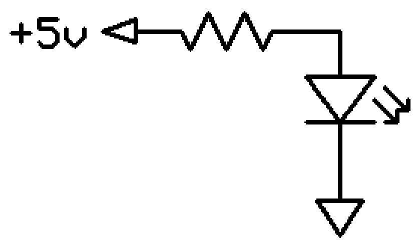

Figure 2 – Simple LED Circuit

The circuit shows how to connect a resistor and an LED to a 5 volt supply and ground. The question is, how large should the resistor be? If it is too large, not enough current will flow and the LED will be very dim; if it is too small, too much current will flow and the LED may burn out. The calculation of the resistor value involves simple circuit analysis and the use of Ohm's Law. In Figure 2, a voltage of 5 volts is applied across the resistor/LED combination. The applied voltage will be divided between the resistor and the LED. That is to say, if you put the positive lead of a voltmeter on the resistor/diode connection and the negative lead of the voltmeter on the ground connection, the meter would indicate a voltage of less than 5 volts – perhaps 2 volts or so. The remaining voltage (5 volts minus 2 volts, in this case) is said to be "dropped" across the resistor. This means that if you moved the positive voltmeter lead to the +5 supply and the negative lead to the resistor/LED connection, the meter would read 3 volts. This observation is useful because a resistor behaves according to Ohm's law which says that the voltage across a resistor is equal to the current flowing through it (in amps) multiplied by its resistance (in ohms). The formula that expresses this concept is $V = I * R$.

To show how to apply this information, we need the specifications for a real LED. For this example, we'll use the Digi-Key part MV50154-ND, a red LED about 5mm in diameter. The specifications for this LED indicate that with approximately 10mA of "forward current" flowing through the LED there will be about 1.6 volts of "forward voltage" across the LED. This leaves 3.4 volts to be dropped across the resistor with the same 10mA of current flowing through it. According to Ohm's Law, the resistor should have the value 3.4 volts divided by 0.01 amps, or 340 ohms. A good choice, then, would be to use the commonly available value of 330 ohms.

Unlike a resistor, the forward voltage across an LED doesn't change much as the current varies over a narrow range. For the purposes of this analysis, we can assume that the forward voltage will be a constant 1.6 volts and, consequently, the voltage across the resistor will remain constant at 3.4 volts. Since we chose a slightly smaller resistor, the resulting current will be slightly higher than the target value that we initially chose. It is simple to calculate the new current as 3.4/330 or 10.3mA. You can modify the resistor value to increase or decrease the current in order to increase or decrease the light output of the LED. However, you must be careful to not to exceed the maximum forward current specification of the LED.

Now that we have established the basic framework for illuminating an LED, how do you connect an LED to a microcontroller so that it can be turned on and off under program control? There are two basic ways that an LED can be connected, both shown in Figure 3.

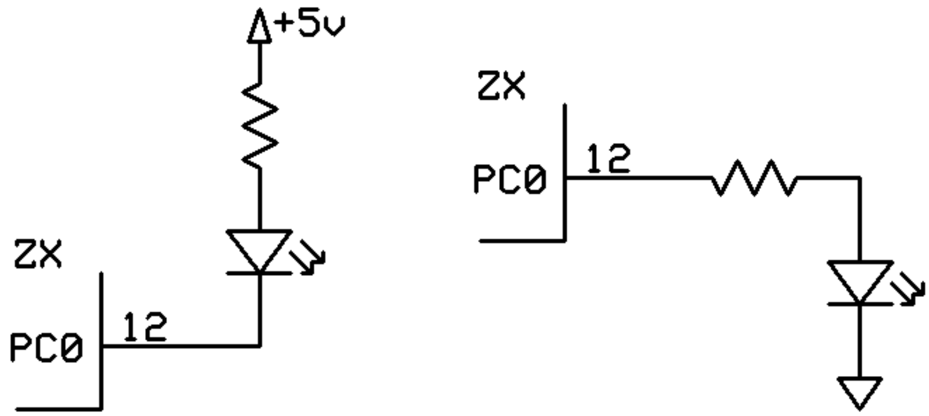

Figure 3 – Controlling an LED with a ZX Output

Both of these circuits work well and which one you choose depends on personal preference and a few other factors. In the circuit on the left of Figure 3, the ZX pin must be set to output logic zero in order for the LED to illuminate. When the ZX pin is in the output high state or the input state very little current will flow through the LED – for practical purposes, the current will be zero. In this configuration, the ZX output is said to "sink" the LED current because current flows from the +5 supply through the resistor and LED and then into the ZX pin to ground.

In the circuit on the right of Figure 3, the ZX pin must be set to output logic one in order for the LED to illuminate. In this configuration, the ZX pin is said to "source" the LED current because current flows from the power supply, into the ZX chip, out though the pin, through the resistor and LED to ground.

One consideration for the ZX-24 is that if the on-board regulator is being used, the LED current in the second configuration has to be supplied by that regulator. The regulator is capable of supplying a total of 100mA of which 50 to 60mA is used by the CPU and other on-board components. The balance is available to by sourced by the output pins.

Connecting a Switch

Switches are useful devices for creating input signals to a microcontroller. Figure 4 shows two different schematic symbols for a type of switch that is called single pole, single throw, or SPST. The "single pole" portion of the description means that there is only one circuit connection that can be made through the switch. The "single throw" portion of the description means that the switch has two positions and a connection is made only in one of them. Switches are available with multiple poles and multiple "throws" but the SPST switch is best suited for this example.

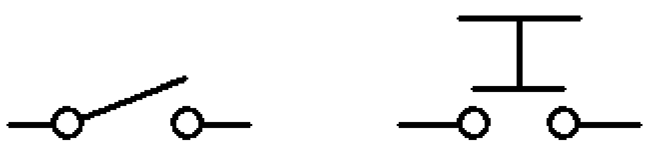

Figure 4 – Schematic Symbols for an SPST Switch

The difference between the two symbols shown in Figure 4 is that the one on the left represents a "toggle" switch that will remain in either the open or the closed position. The typical wall light switch in your home is an example of a toggle switch. The symbol on the right is a "momentary" switch that is normally open but that will make contact when you push the button. When you release the button it returns to the open position. A typical doorbell button is an example of a momentary switch. (There are also momentary switches that are normally closed. In that case, pressing the switch button opens the switch and it returns to the closed position when released.)

A switch can be connected to a microcontroller in two ways – either to present a logic one when closed or to present a logic zero when closed. Again, which method is used depends largely on preference. The schematic of Figure 5 shows both configurations.

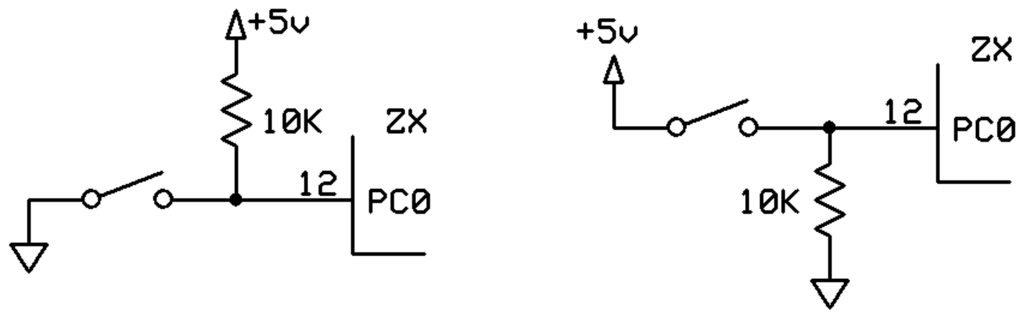

Figure 5 – Simplified Connection of an SPST Switch

In the circuit on the left, we assume that the pin has been made an input. While the switch is open, invoking GetPin(12) will return the value 1 because of the effect of the pullup resistor. If you call GetPin(12) when the switch is closed, the return value will be 0. Note that if the I/O pin is configured as zxInputPullup, the external pullup resistor is not required.

In the circuit on the right, we again assume that the pin has been made an input. While the switch is open, invoking GetPin(12) will return the value 0 because of the presence of the pulldown resistor. If you call GetPin(12) when the switch is closed, the return value will be 1. Note that if the I/O pin is configured as zxInputPullup, the internal pullup resistor forms a voltage divider with the external pulldown resistor. Since the internal pullup resistor is in the range of 20K to 50K ohms, the voltage level at the I/O pin with the switch open will be somewhere between 1.7 volts and 0.8 volts. This is not acceptable since voltages between 1.0 and 3.0 volts are “indeterminate” – they might be read as a zero or they might be read as a one. Using a smaller pulldown resistor may improve the situation but it is best to not configure the I/O pin with the pullup active when using the circuit on the right.

The value of the pullup and pulldown resistors is not particularly critical. Any value from 1K to 100K will probably work fine. Lower values will help prevent electrical “noise” from affecting the result at the expense of increasing the power consumption of your circuit. 10K is a good compromise between these factors.

There is a small risk of damaging your microcontroller when using the circuits as shown in Figure 5. Consider what would happen in the lefthand circuit if the I/O pin were inadvertently configured to zxOutputHigh and the switch were closed. In that circumstance, the output driver is being shorted to ground and there is a possibility that the driver for that I/O pin will burn out. A similar situation would prevail in the righthand circuit if the pin were inadvertently configured as zxOutputLow; the power supply will be shorted through the pin driver to ground. To protect the I/O pins from this fate, it is highly recommended to always include a small resistance in series with the I/O pin as shown in the modified schematics of Figure 6. The 1K resistors serve to limit the current that would flow in the adverse conditions described above.

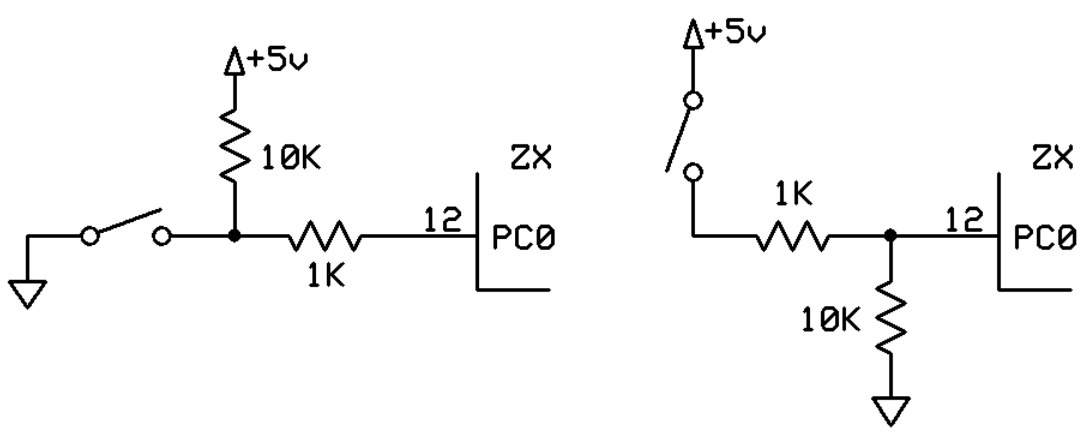

Figure 6 – Connection of an SPST Switch with Limiting Resistors

Dealing with Contact Bounce

With any mechanical switch, the contacts rarely open and close cleanly. Rather, when you close a switch, the contacts are driven together and when they make contact, they bounce back apart, close again, bounce apart again, etc. Eventually, the bouncing ceases and the contacts remain closed. A similar phenomenon occurs when you open the contacts.

The duration of the contact bounce depends on the type of switch, the quality of the materials and workmanship, the condition of the contacts and other factors but a good rule of thumb is that the duration will be in the range of 30mS to 50mS. Due to the speed of the ZX processors, it is possible for the code to detect a change in the state of the switch, respond to the change and then read the state of the switch again before it has ceased bouncing. Depending on the timing, this often produces undesirable results. For example, say your program is keeps track of how many times a pushbutton switch closes and increments a counter. The false closures due to contact bounce will likely lead to erroneous counts.

So, what can be done about contact bounce? The methods employed to deal with this phenomenon are called de-bouncing and it can be done in hardware or in software. To do so in software is fairly simple: sample the switch state and when a change is detected, delay for 30mS to 50mS before proceeding. That simple delay helps to ensure that your program won't get back around to testing the switch state again until after the bouncing has ceased.
The software debounce method works well for many situations but for some it does not. For example, if the flow of the software cannot tolerate the required delay it may be necessary to implement the debouncing in hardware. The schematic of Figure 7 shows a simple switch debouncing circuit.

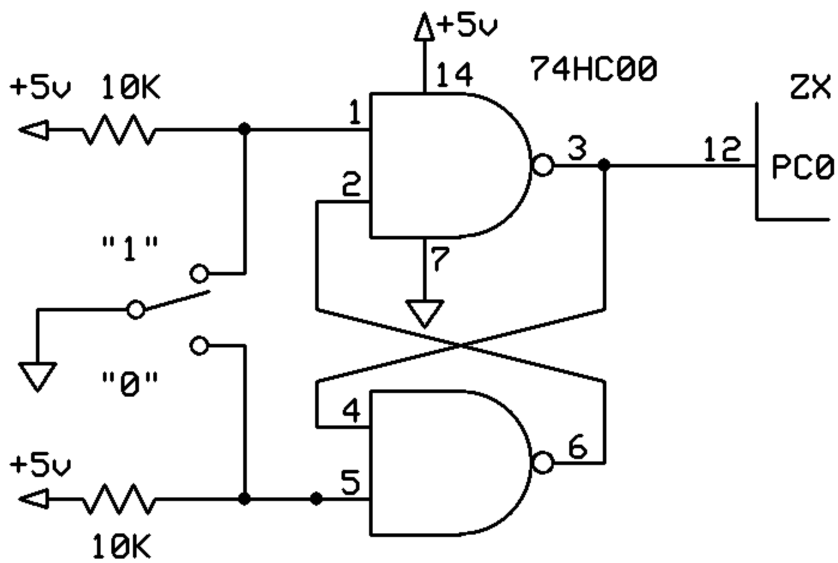

Figure 7 – Switch with Hardware Debounce

This circuit employs a single pole, double throw (SPDT) switch and two NAND gates connected as an S-R flip-flop. The switch's two positions are labeled relative to the logic signal generated by the upper NAND gate. Of course, the lower NAND gate's output will be the complement (opposite logic level) of that of the upper gate.

Connecting an Output Signal from a Device

Occasionally, you'll want to connect a device to your ZX that produces a digital output signal whose voltage excursions are not directly compatible with the ZX inputs. Depending on the voltage levels involved, the solution may be as simple as employing clamping diodes and a limiting resistor as shown in Figure 8. The SIG label represents a signal from an external device.

If the range of the device output is zero to 5 volts, the clamping diodes are not required. They are sometimes used as a safety measure even when the voltage swing is compatible. This is especially true if the device is connected by long leads or if it will operate in an electrically "noisy" environment.

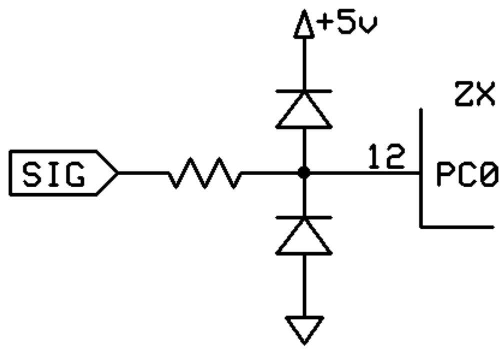

Figure 8 – Using Clamping Diodes

If the output signal goes above 5 volts or below ground, the clamping diodes prevent the ZX from being exposed to voltage levels that might damage the input circuitry. The value of the limiting resistor is calculated to limit the current flow to safe levels when the signal goes above +5 or below ground. The diodes are selected to be compatible with the resulting current flow and the voltage levels that are present. A commonly used diode is the 1N914B or 1N4148.

As an example, say that the device produces a signal that alternates between +15 volts and -15 volts. When the signal is at +15 volts, the ZX input pin will rise to about 5.6 volts at which point the upper diode will begin conducting and clamp the voltage at that level. This means that there will be 9.4 volts across the limiting resistor. On the other hand, when the signal goes to –15 volts, the input pin will drop to –0.6 volts and the lower diode will begin conducting, clamping the voltage at that level. In this case, there will be 14.4 volts dropped across the resistor. Since this is the larger of the two voltage levels, the limiting resistor needs to be made large enough so that the current is limited to a safe level in that case. For example, if the device can supply up to 20mA of current, the minimum resistor value would be 14.4/0.02 = 720 ohms. A larger value, perhaps 1K to 47K can be used to limit the current flow even further.

When connecting any device output to the ZX, you must be aware of the voltage levels that will be properly interpreted by the ZX as logic zero and logic one. The specifications of the ZX are that a voltage less than 0.2 times Vcc will be interpreted as a logic zero while a voltage above 0.6 times Vcc will be interpreted as a logic one. When running the ZX at 5 volts, these levels are 1.0 volts and 3.0 volts, respectively. The range between these levels is the “indeterminate” zone where there is no guarantee how the ZX will interpret the input.

Consider another hypothetical device whose output alternates between zero and +2 volts. In this case, the upper voltage is not high enough to be guaranteed to be interpreted as logic 1 so we need some circuitry to condition the signal. A commonly used technique for this situation is to employ a transistor as a switch as shown in Figure 9.

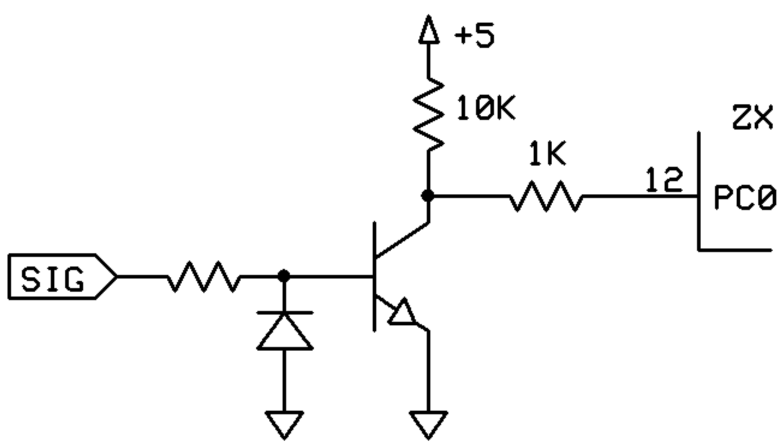

Figure 9 – Employing a Transistor Switch

This circuit converts the smaller voltage swing of the output device to a 0 to 5 volt swing that is compatible with the ZX input. It also inverts the logic signal. In order to explain how this works and how the component values are selected, we need to first present some fundamentals of transistor operation.

Transistor Switch Fundamentals

A transistor is a current amplifier that has three terminals called the base, the collector and the emitter. There are two basic types of transistors – NPN and PNP. The circuit of Figure 9 employs an NPN transistor with the emitter connected to ground, the device's output signal driving the base through a limiting resistor and the collector connected to the ZX input pin through a limiting resistor and pulled up to the supply voltage. The clamping diode connected to the base is only necessary if the signal goes significantly below zero volts so it is superfluous in this specific case since the device signal is assumed to be 0 to 2 volts.

A transistor has three "regions" of operation called cutoff, saturation and the linear region. The linear region is avoided when operating the transistor as a switch so it is mentioned here only for completeness. When operating in the cutoff region, the path from collector to emitter exhibits a high resistance so very little current flows and the collector-emitter voltage is essentially the supply voltage. When operating in the saturation region, the path from collector to emitter exhibits a very low resistance and the collector-emitter voltage is very small (typically 0.1 volts or so). From this description you can see how these two regions of operation can be used to implement a switch - saturation is equivalent to "on" and cutoff is equivalent to "off".

So, how do you cause the transistor to operate at cutoff or saturation? Cutoff is the simpler of the two so we'll address it first. But first, we must introduce the concept of Beta. One of the characteristics of a transistor is that, over a significant portion of the operating range, the collector current is equal to the base current times a gain factor, called Beta or h_{fe}. The value of Beta is given on the transistor's datasheet, sometimes as a graph showing how the Beta factor is different at various collector current levels. Beta also varies with temperature and other factors but within a narrow range of current we can make the simplifying assumption that it is a constant value. For the commonly used 2N3904 transistor, a typical Beta value is 100 at a collector current of 1mA. A datasheet for the 2N3904 may be found at http://www.fairchildsemi.com/ds/MM%2FMMBT3904.pdf.

To force the transistor to the cutoff region it is necessary to ensure that very little, ideally zero, current flows into the base of the transistor. Referring back to Figure 9, it is easy to see that if the input signal to the transistor is zero volts then no current will flow into the base of the transistor and, therefore, no collector current will flow. This means that the transistor is operating in the cutoff region and the voltage at the collector will be nominally 5 volts.

To force the transistor to the saturation region when the input signal is at +2 volts, we need to choose a value for the base resistor that will cause sufficient current to flow that, when multiplied by the Beta factor to yield the collector current, will result in the collector dropping to nearly zero volts. We can compute this value by working backwards. Firstly, we must determine the collector current that would result in the collector-emitter voltage being nearly zero. Using Ohm's law, we can easily compute that as 5 volts divided by 10K ohms or 0.5mA. Then, assuming a Beta of 100 we determine that a base current of, at a minimum, 0.005mA will be required. In order to ensure that the transistor will indeed be in saturation we can over-drive it, say by a factor of 10, by arranging for 0.050mA of base current.

Before calculating the value of the base current limiting resistor, we need one final transistor parameter: the base-emitter voltage. As an approximation, the base-emitter junction can be thought of as being just a diode (anode = base, cathode = emitter for an NPN transistor). Using this simplification, we can make the assumption that the base-emitter voltage will not rise above about 0.6 volts. This allows us, then, to calculate the required base resistor as (2 volts – 0.6 volts) divided by 0.050 mA = 28K ohms. The nearest standard value is 27K but 22K or 33K would work equally well.

Using an Optoisolator

Another useful device for connecting signals to a ZX is an optoisolator. An optoisolator is very similar to the transistor switch just discussed with the difference that instead of driving the base directly with current, an LED emits photons that cause the transistor to turn on. This technique has the advantage that there needn't be a common ground between the signal source and the receiving circuitry. An example circuit utilizing an optoisolator is shown in Figure 10.

The LED in an optoisolator is usually called the photoemitter or simply the emitter. The transistor is called a photodetector or detector. An optoisolator exhibits a characteristic similar to a transistor's Beta factor but it is called Current Transfer Ratio. The CTR expresses the ratio of the detector's collector current to the emitter's forward current flow. The datasheet for the 4N25 (http://www.fairchildsemi.com/ds/4N/4N25.pdf) indicates that the CTR is about 20 over the range of forward emitter (LED) current of 2 to 10mA. However, the CTR falls off rapidly below 2mA, less so above 10mA.

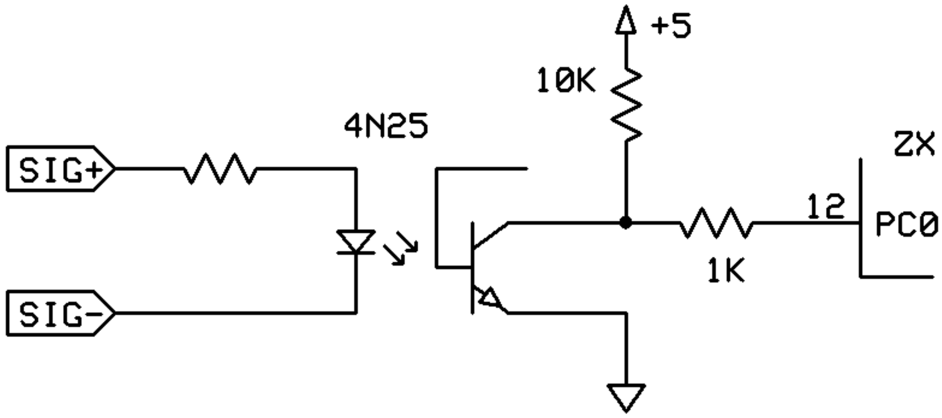

Figure 10 – Connecting an Input Using an Optoisolator

Working backwards again we can compute that we need to ensure that 0.5mA of collector current will flow so that the collector drops to zero volts. Dividing this by the CTR of 20 indicates that at least 0.025mA of forward current will be required through the LED. To be certain, we'll arrange for a forward current of 2mA since the CTR drops so fast below that level. Consulting the datasheet again, we find that the forward voltage across the LED with 2mA of forward current will be about 1.1 volts. Using these values plus the magnitude of the signal, the size of the limiting resistor can be calculated using Ohm's law. If, for example, the amplitude of the signal is 5 volts, the limiting resistor should be (5 volts - 1.1 volts) / 2mA = 4.95K ohms. A standard value of 4.7K or 5.1K would be appropriate.

An optoisolator can also be used for signals being sent to a device from the ZX. This would be used primarily in situations where it is necessary or desirable to avoid having a common ground between the ZX and the device.

Driving a Load

The earlier section in this application note that showed how to drive an LED presented principles that are applicable to driving many different types of loads. A more general diagram, showing a generic load device, is shown in Figure 11.

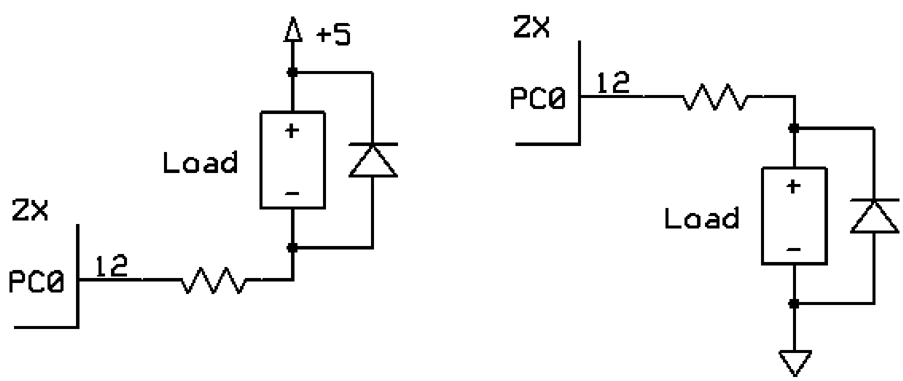

Figure 11 – Driving a Generic Load

Note the similarity between these diagrams and those shown in Figure 3. The difference is that the LED has been replaced by a generic load device that could be an LED, a solenoid, a relay, a lamp or any other device that is compatible with a 5 volt source. The extra diode across the load device is called a “snubber” and its purpose is to suppress harmful voltages that are generated by “inductive kick back”. If the load is non-inductive, the snubber is not required. Examples of inductive loads include solenoids, relays and transformers. Generally, any load that contains a coil of wire will exhibit inductive characteristics. If in doubt, add the snubber; it will never be harmful.

If the load device is specified to operate directly on 5 volts, the limiting resistor shown in Figure 11 is probably not required. Otherwise, the calculation of the value of the limiting resistor is performed in exactly the same way as described earlier for the LED. Determine the required current limit, determine the voltage drop across the load at that current flow and use Ohm's law to calculate the resistance value to drop the remainder of the voltage at that current flow.

  Published January 2006

If the load device requires more current or voltage than can be handled by the ZX output pin then the load cannot be driven directly. In this case, an intermediary driver circuit will be required, a simple example of which is shown in Figure 12. Here again, the snubber diodes are not required unless the load is inductive.

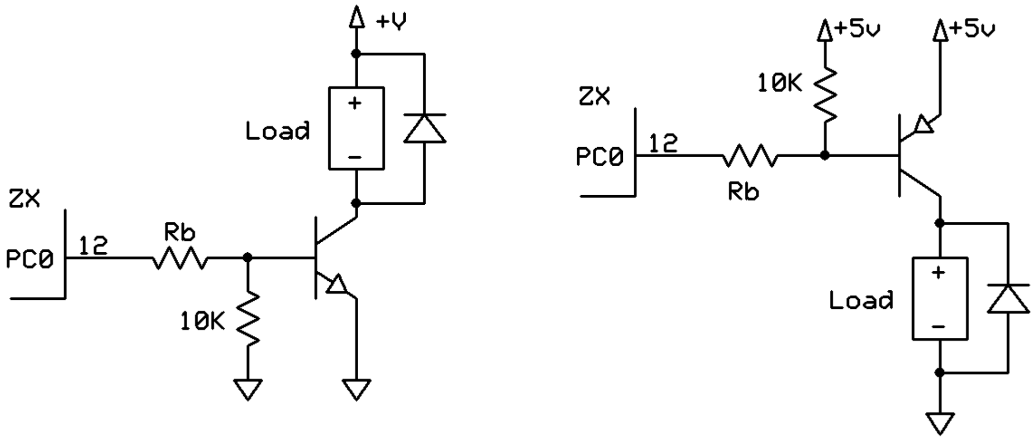

Figure 12 – Driving a Load Using a Transistor

The diagram on the left side of Figure 12 shows a load being switched by an NPN transistor. The transistor is turned on by setting the ZX pin to a logic one level and the value of Rb is calculated using the concepts presented in the earlier section on Transistor Switch Fundamentals. Commonly used NPN transistors are the 2N3904 and 2N2222A but there are many others that are designed to be used in general purpose switching applications like this. Transistors have a maximum collector current specification than should not be exceeded. The limit is 200mA for the 2N3904 and 600mA for the 2N2222A. Other transistors have much higher current ratings, e.g., the TIP120 (http://www.fairchildsemi.com/ds/TI%2FTIP120.pdf) which can handle 5 amps of current and has a snubber diode built in. There are also available arrays of driver transistors that are convenient to use when multiple drivers are required. One example of a driver array is the ULN2803 (http://www.toshiba.com/taec/components/Datasheet/ULN2803AP.pdf) that provides 8 high current drivers in a single package. (Both the TIP120 and the ULN2803 incorporate what is known as a Darlington configuration to achieve higher gain – see http://en.wikipedia.org/wiki/Darlington_transistor.)

The schematic on the right side of Figure 12 illustrates the use of a PNP transistor. This configuration is useful if one side of the load must be grounded. Note, however, that with the PNP driver the ZX output must be set to logic zero to turn on the transistor and current will be drawn out of the base of the transistor to cause current to flow out of the collector into the load. Other than the reversed currents and voltages the fundamentals described for the NPN transistor apply to the PNP as well. Commonly used PNP transistors are the 2N3906 and 2N2907A. These two transistors are said to be complementary to the 2N3904 and 2N2222A, respectively because they have similar characteristics but are of the opposite polarity.

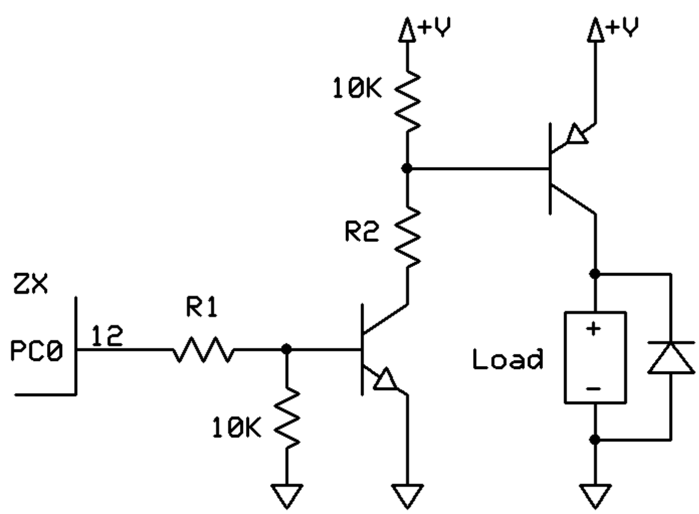

Figure 13 – Driving a High Voltage Load Using a PNP Transistor

One important consideration when driving a load with a PNP transistor is that the supply voltage for the load must be 5 volts as shown in Figure 12 in order to be compatible with the ZX output. (This is not the case with an NPN driver, however.) If you need to use a higher voltage source and you also need to have one side of the load grounded, you must use a multi-stage driver, an example of which is shown in Figure 13. In the multi-stage circuit, an NPN transistor is added between the ZX output pin and the PNP driver. To energize the load, the ZX output pin is set high to switch on the NPN transistor. This causes current to flow into the NPN's collector, drawing current from the PNP's base switching it on as well. Again, the same procedure as presented earlier is used to calculate the resistances needed to generate the required current flows.

The ideas presented in this section also apply when using a mechanical or solid state relay to control a load that requires high voltage, high current or is AC powered. In these cases, the relay's control input is the load being driven by the ZX.

Interfacing to Lower Voltage Devices

It is becoming more common to encounter devices that operate on less than 5 volts, e.g. 3.3 volts. Such devices have logic outputs that swing from zero to their supply voltage and are therefore not directly compatible with a ZX running at 5 volts. Depending on the device, its inputs may be 5-volt tolerant but if not, you can add level conversion circuitry allow connection between the ZX and low voltage devices. There are commercially available bidirectional multi-channel logic level translators available such as the MAX13101E. These are most useful when a large number of signals need to be converted. If your application has only a few signals that require conversions there are some simpler, perhaps less expensive, alternatives.

The circuits in Figure 14 illustrate a simple way of making a logic conversion. Although the configuration may look odd, its operation is quite simple. Consider the circuit on the left of Figure 14. This configuration allows a 3.3 volt output to drive a ZX input. When the device output is high (near 3.3 volts) the voltage at the ZX input will rise to about 0.6 volts above the device output or about 3.9 volts. This is high enough to be recognized as a logic one by the ZX. When the device output is low (near zero volts) the ZX input will be about 0.6 volts above that level, low enough to be recognized as a logic zero by the ZX. The circuit on the right side of Figure 14 operates on a similar principle but the diode blocks the 5-volt output from the ZX from reaching the lower voltage input.

It should be pointed out that the pullup resistor in the circuit on the left of Figure 14 is not necessary if the input pullup is enabled. The same may apply to the circuit on the right if the external 3.3V device has a pullup on its input.

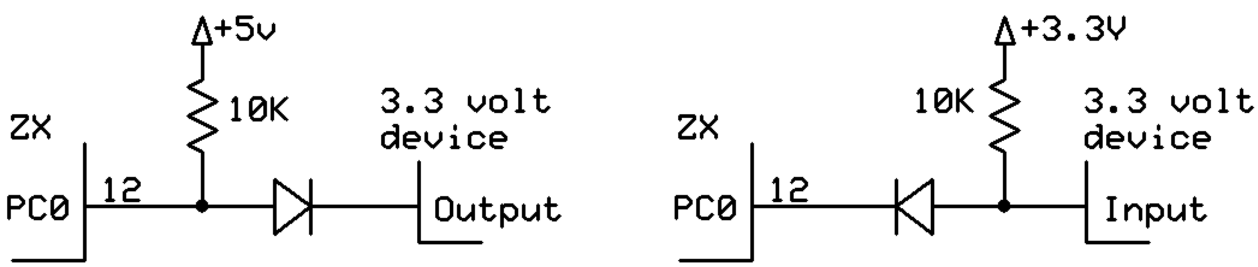

Figure 14 – Logic Level Translation Using Diodes

One issue with this type of circuit is the voltage drop across the diode. A common diode like the 1N914B has a forward voltage drop of about 0.6 volts although it does vary with the forward current. This forward drop adds to the output low voltage of the driving pin providing a logic zero voltage at the input of perhaps 0.8 to 0.9 volts. For the ZX input, this will be properly recognized as a logic zero. You'll have to check the datasheet for you low voltage device to determine if that is an acceptable logic zero voltage or not. If the forward voltage of the diode is problematic, a different diode may be selected, e.g. a germanium or Schottky diode, that has a lower forward voltage.

Other alternatives for unidirectional logic level translation are shown in Figures 15 and 16. The circuits of Figure 15 are similar to the diode circuits except that transistors are used instead. The advantage of this is that the collector-emitter voltage of a transistor is very low – on the order of 0.2 volts or so depending on the collector current. This lower voltage results in a much lower logic zero level at the input of the device than can be achieved using diodes, thus providing better noise immunity.

In the circuit on the left side of Figure 15 an output from a 3.3 volt device is connected to a ZX pin configured to be an input. When the device's output is at logic zero, the transistor turns on dropping the collector to nearly the same output level. When the device's output is at logic one, the transistor turns off and the collector is pulled up to +5 by the resistor. Activating the ZX pin's internal pullup resistor may eliminate the need for the external pullup. The circuit on the right side of Figure 15 works by the same principle in the opposite direction.

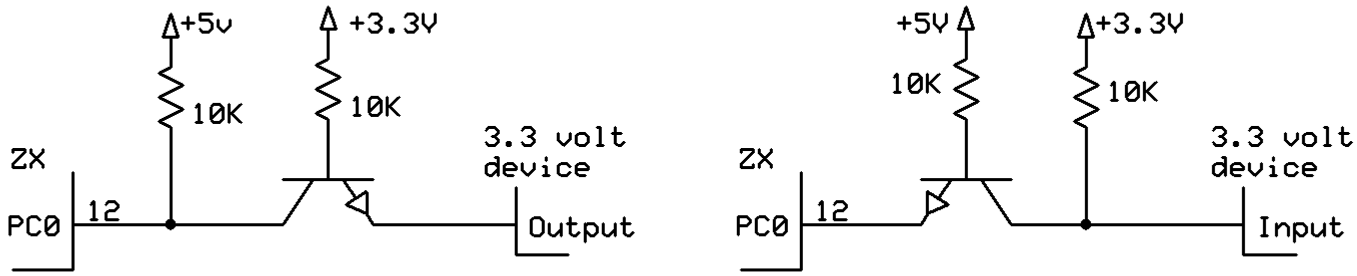

Figure 15 – Logic Level Translation Using Transistors

The circuits of Figure 16 use the same device in two different technologies (HCT and AHC) for the two level translations. On the left, the HCT technology is used because it operates on 5 volts and will properly recognize the output voltages from the 3.3 volt device. The AHC technology is used on the right because it can operate on 3.3 volts and has 5-volt tolerant inputs.

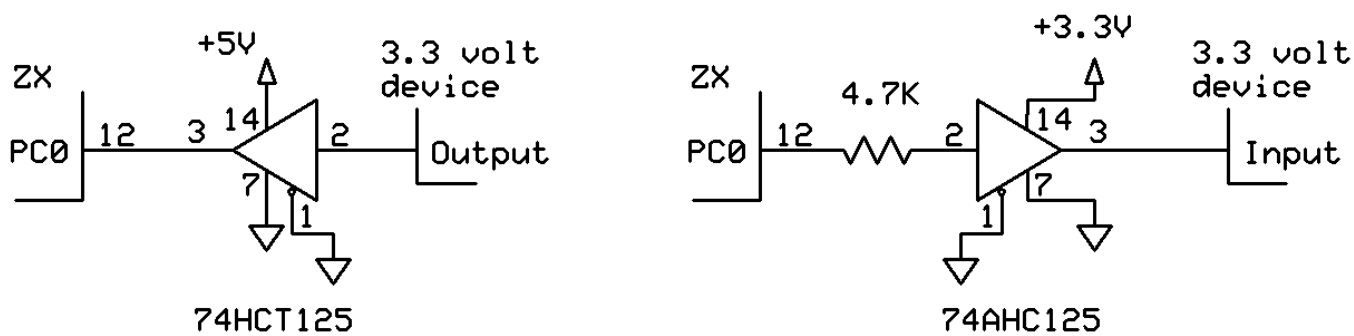

Figure 16 – Logic Level Translation Using Buffer Gates

Summary

Many different kinds of devices can be safely connected to a ZX by applying a few simple principles. The key is to carefully read the datasheet of any device that you intend to use to determine the current and voltage characteristics. Once you have those in hand the interface circuitry, if required, can be selected and component values calculated.

Author

Don Kinzer is the founder and CEO of Elba Corporation. He has extensive experience in both hardware and software aspects of microprocessors, microcontrollers and general-purpose computers. Don can be contacted via email at *dkinzer@zbasic.net*.