

# OPTOCOUPPLERS: WHEN & HOW TO USE THEM

There are many situations where signals and data need to be transferred from one subsystem to another within a piece of electronics equipment, or from one piece of equipment to another, without making a direct 'ohmic' electrical connection. Often this is because the source and destination are (or may be at times) at very different voltage levels, like a microprocessor which is operating from 5V DC but being used to control a triac which is switching 240V AC. In such situations the link between the two must be an *isolated* one, to protect the microprocessor from overvoltage damage.

Relays can of course provide this kind of isolation, but even small relays tend to be fairly bulky compared with ICs and many of today's other miniature circuit components. Because they're electro-mechanical, relays are also not as reliable — and only capable of relatively low speed operation. Where small size, higher speed and greater reliability are important, a much better alternative is to use an *optocoupler*. These use a beam of light to transmit the signals or data across an electrical barrier, and achieve excellent isolation.

Optocouplers typically come in a small 6-pin or 8-pin IC package, but are essentially a combination of two distinct devices: an optical *transmitter*, typically a gallium arsenide LED (light-emitting diode) and an optical *receiver* such as a phototransistor or light-triggered diac. The two are separated by a transparent barrier which blocks any electrical current flow between the two, but does allow the passage of light. The basic idea is shown in Fig.1, along with the usual circuit symbol for an optocoupler.

Usually the electrical connections to the LED section are brought out to the pins on one side of the package and those for the phototransistor or diac to the other side, to physically separate them as much as possible. This usually allows optocouplers to withstand voltages of anywhere between 500V and 7500V between input and output.

Optocouplers are essentially digital or switching devices, so they're best for transferring either on-off control signals or digital data. Analog signals can be transferred by means of frequency or pulse-width modulation.

## Key Parameters

The most important parameter for most optocouplers is their transfer efficiency, usually measured in terms of their

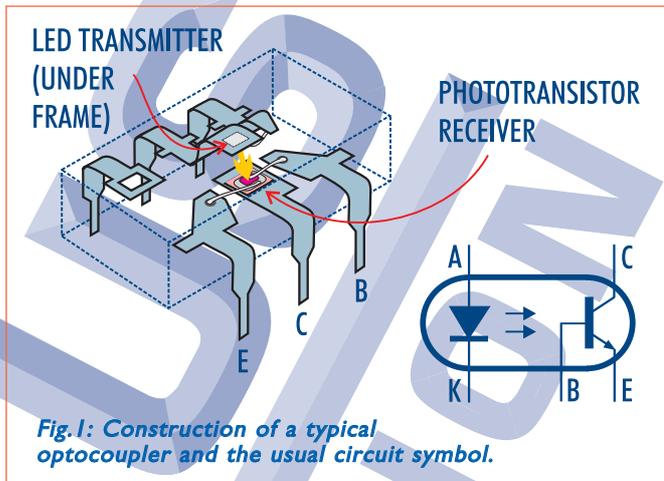


Fig.1: Construction of a typical optocoupler and the usual circuit symbol.

*current transfer ratio* or **CTR**. This is simply the ratio between a current change in the output transistor and the current change in the input LED which produced it. Typical values for CTR range from 10% to 50% for devices with an output phototransistor and up to 2000% or so for those with a Darlington transistor pair in the output. Note, however that in most devices CTR tends to vary with absolute current level. Typically it peaks at a LED current level of about 10mA, and falls away at both higher and lower current levels.

Other optocoupler parameters include the output transistor's maximum collector-emitter voltage rating **VCE(max)**, which limits the supply voltage in the output circuit; the input LED's maximum current rating **IF(max)**, which is used to calculate the minimum value for its series resistor; and the optocoupler's **bandwidth**, which determines the highest signal frequency that can be transferred through it — determined mainly by internal device construction and the performance of the output phototransistor. Typical opto-couplers with a single output phototransistor may have a bandwidth of 200 - 300kHz, while those with a Darlington pair are usually about 10 times lower, at around 20 - 30kHz.

## How They're Used

Basically the simplest way to visualise an optocoupler is in terms of its two main components: the input LED and the output transistor or diac. As the two are electrically isolated, this gives a fair amount of flexibility when it comes to connecting them into circuit. All we really have to do is work out a convenient way of turning the input LED on and off, and using the resulting switching of the photo-transistor/diac to generate an output waveform or logic

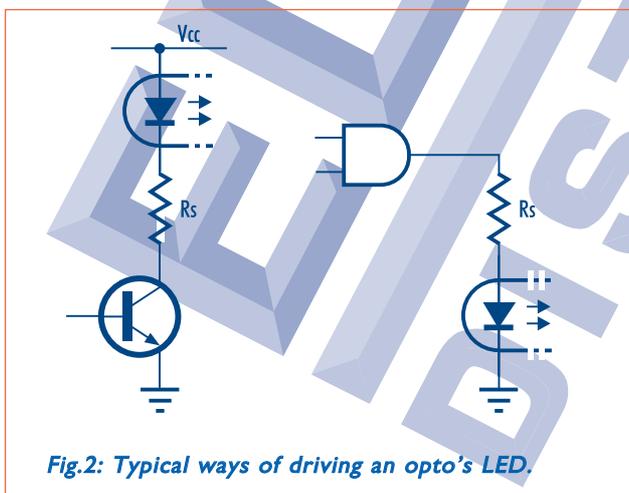


Fig.2: Typical ways of driving an opto's LED.

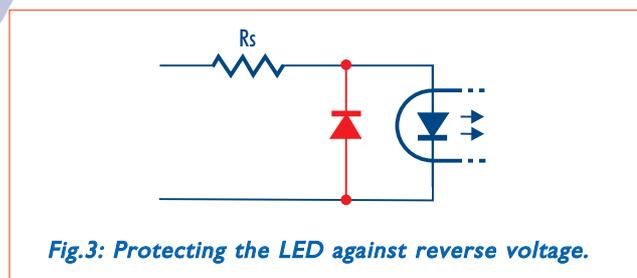


Fig.3: Protecting the LED against reverse voltage.

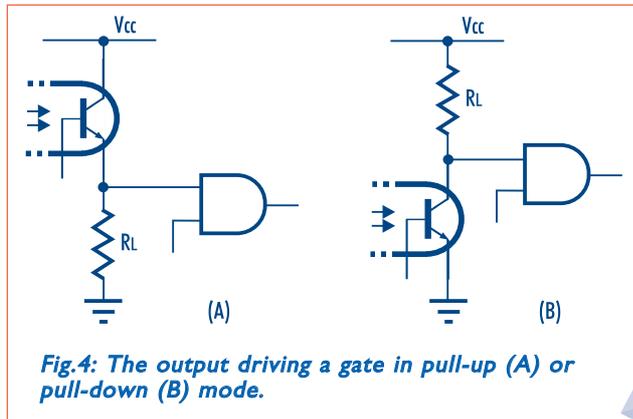


Fig. 4: The output driving a gate in pull-up (A) or pull-down (B) mode.

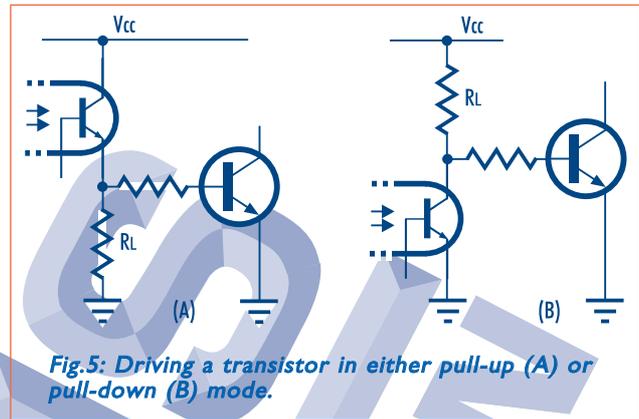


Fig. 5: Driving a transistor in either pull-up (A) or pull-down (B) mode.

signal that is compatible with our output circuitry.

For example just like a discrete LED, you can drive an optocoupler's input LED from a transistor or logic gate/buffer. All that's needed is a series resistor to set the current level when the LED is turned on. And regardless of whether you use a transistor or logic buffer to drive the LED, you still have the option of driving it in 'pull down' or 'pull up' mode — see Fig.2. This means you can arrange for the LED, and hence the optocoupler, to be either 'on' or 'off' for a logic high (or low) in the driving circuitry.

In some circuits, there may be a chance that at times the driving voltage fed to the input LED could have reversed polarity (due to a swapped cable connection, for example). This can cause damage to the device, because optocoupler LEDs tend to have quite a low reverse voltage rating: typically only 3 - 5V. So if this is a possibility, a reversed-polarity diode should be connected directly across the LED as shown in Fig.3.

On the output side, there are again a number of possible connections even with a typical optocoupler of the type having a single phototransistor receiver (such as the 4N25 or 4N28). In most cases the transistor is simply connected as a light-operated switch, in series with a load resistor  $R_L$  (see Fig.4). The base of the transistor is left unconnected, and the choice is between having the transistor at the top of the load resistor (Fig.4A) or at the bottom (Fig.4B) — i.e., in either pull-up or pull-down mode. This again gives plenty of flexibility for driving either logic gates or transistors, as shown in Fig.5.

If a higher bandwidth is needed, this can be achieved by using only the collector and base connections, and using the transistor as a photodiode (Fig.6A). This lowers the optocoupler's CTR and transfer gain considerably, but can increase the bandwidth to 30MHz or so. An alternative

approach is still to use the output device as a photo-transistor, but tie the base down to ground (or the emitter) via a resistor  $R_b$ , to assist in removal of stored charge (Fig.6B). This can extend the opto's bandwidth usefully (although not dramatically), without lowering the CTR and transfer gain any more than is necessary. Typically you'd start with a resistor value of 1M $\Omega$ , and reduce it gradually down to about 47k $\Omega$  to see if the desired bandwidth can be reached.

A variation on the standard optocoupler with a single output phototransistor is the type having a photo-Darlington transistor pair in the output (Fig.7), such as the 6N138. As mentioned earlier this type of device gives a much higher CTR and transfer gain, but with a significant penalty in terms of bandwidth. Connecting a base tieback resistor as in Fig.7B can again allow a useful extension of bandwidth without sacrificing too much in terms of transfer gain.

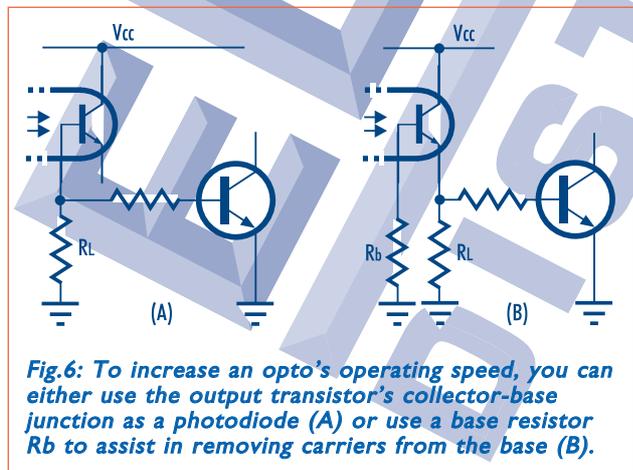


Fig. 6: To increase an opto's operating speed, you can either use the output transistor's collector-base junction as a photodiode (A) or use a base resistor  $R_b$  to assist in removing carriers from the base (B).

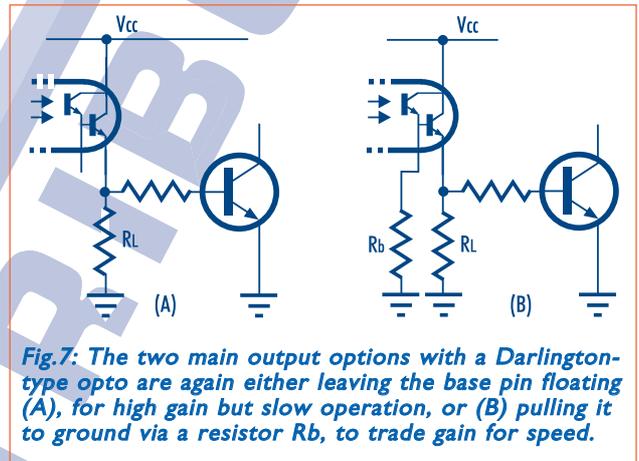


Fig. 7: The two main output options with a Darlington-type opto are again either leaving the base pin floating (A), for high gain but slow operation, or (B) pulling it to ground via a resistor  $R_b$ , to trade gain for speed.

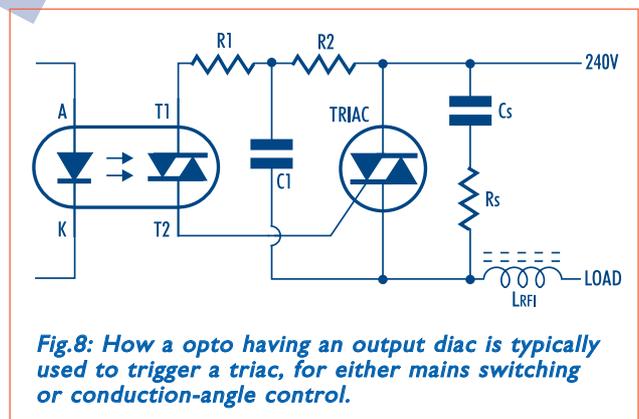


Fig. 8: How a opto having an output diac is typically used to trigger a triac, for either mains switching or conduction-angle control.

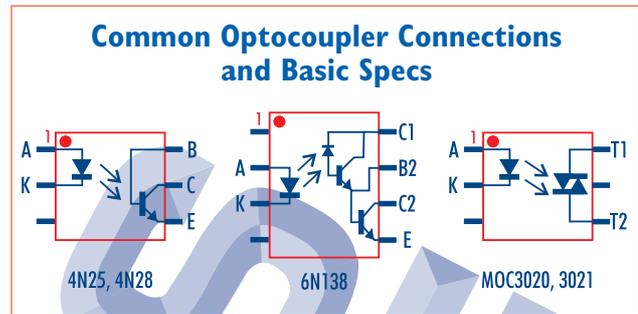
## Electus Distribution Reference Data Sheet: OPTOCOUP.PDF (3)

The other main type of optocoupler you'll tend to encounter is the type having an output diac or bilateral switch, and intended for use in driving a triac or SCR. Examples of these are the MOC3020 and MOC3021. Here the output side of the optocoupler is designed to be connected directly into the triggering circuit of the triac (Fig.8), where it's operating from and floating at full AC mains potential.

As you'd expect the output diac is connected into the triac gate triggering circuit in much the same way as a discrete diac. You need a filter/delay circuit before the diac (R1-2 and C1) and the usual snubber circuit across the triac (Rs, Cs) to ensure correct triggering with inductive loads. Normally you'd also need at least an RFI suppressor choke LRFI as well, plus a suitable capacitor across the load.

Basic performance specs for the optocouplers stocked by Electus Distribution are shown in the table at right, while their pin connections are shown at top right.

**(Copyright © 2001, Electus Distribution)**



| TYPE    | ISOLATION (V <sub>iso</sub> ) | INPUT LED I <sub>F(max)</sub> | OUTPUT V <sub>CE(max)</sub> | CTR <sub>min</sub> (@ I <sub>F</sub> ) | BANDWIDTH (kHz) |
|---------|-------------------------------|-------------------------------|-----------------------------|--|-----------------|
| 4N25    | 5300Vrms                      | 80mA                          | 7V                          | 20% (10mA)                             | 300             |
| 4N28    | 5300Vrms                      | 80mA                          | 7V                          | 10% (10mA)                             | 300             |
| 6N138   | 2500Vrms                      | 20mA                          | 7V                          | 300% (1.6mA)                           | ~20             |
| MOC3020 | 7500Vpk                       | 50mA                          | V <sub>off</sub> = 400V     | (Trig. @ 30mA)                         | —               |
| MOC3021 | 7500Vpk                       | 50mA                          | V <sub>off</sub> = 400V     | (Trig. @ 15mA)                         | —               |