Using Switches and Relays for Automatic or Manual Modes

Michael Moulton

April 2, 2010

Executive Summary: Relays are components which allow a low-power circuit to switch a relatively high current on and off, or to control signals that must be electrically isolated from the controlling circuit itself. This application note deals with relays specifically for our project, including their specific use, how to power them, and actions taken in the event of their failure.

Keywords: relays, switches, transistor buffer, communication failure, manual mode, automatic mode

Table of Contents

Introduction..**3**
Objective..**3**
Issues..**4**
Conclusion...**6**
References...**6**

Introduction

Relays are used for a variety of applications; one them being geared specifically towards the functioning of our project, which is that of switching. In order to power specific components of our design, relays will be used to 'relay' a signal from a microcontroller to those devices which need to be turned on or off. When using relays, there is a chance for it to eventually fail. Taking this into account, a scheme should be developed to by-pass the relay path to directly power the component in case of relay failure.

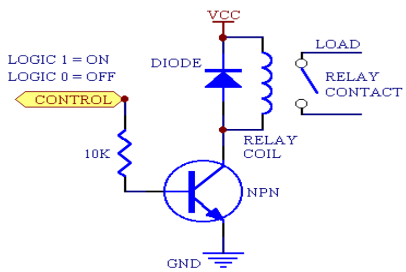
Objective

With our project, we will be using a programming a Peripheral Interface Controller or PIC to send a signal to peripheral components of our design to turn them on. More specifically, those components are Wi-Fi and satellite routers used for internet connection. When the microcontroller receives a signal from our radio link to initiate internet connection for a specific location, the microcontroller will then take appropriate action to turn on the devices necessary to make internet available. One way of doing this is to send the signal from the microcontroller to a relay that will just relay power to that particular device. Basically, microcontroller receives signal to turn on device, microcontroller sends signal to relay, relay activates, then peripheral device (Wi-Fi router) is activated.

Issues and Resolutions

Current – In order to make a relay operate, one must pass suitable “pull-in” and “holding” current through its energizing coil. For our project, we wish to power equipment that runs off of a typical household voltage of 120 VAC, and to power the relay from a 12 VDC supply. As an example, take the S87R from Tyco Electronics. For a 12 VDC supply, the minimum amount of current required to be drawn through the coil for single pole model is about 240 mA. Unfortunately, the PIC microcontroller only outputs about 25 mA. Obviously this is not enough amperage to generate enough magnetic field strength in the coil to switch the relay into its ‘on’ state. One solution is to use a transistor as a buffer.

Transistor Buffer – One way of getting the required amount of current drawn through the coil of the relay is to use a transistor as a buffer.



Example of using a transistor to power a relay

The above picture is an example of how to wire a relay to control a large power supply to a peripheral device via a small supply current from a microcontroller and using a transistor to amplify the supply current to the relay. Of course component values, such as resistors vary according to a variety of factors. For the above picture, this example used 10 kOhms for the value of R1. In practice, one determines the value of R1 in the following manner:

1. Determine the minimum amount of current needed to activate the relay.
2. This is also the current being drawn through the collector of the transistor, keep in mind that we with the transistor to be in the saturation region.
3. Determine the base current necessary to bias the transistor in the saturation region when the collector current is determined.

For our specific project, we have 12 VDC supply and the relay needs 240 mA of current to activate, thus this is also the current through the collector. In order for the transistor to go into saturation (assuming an I_c/I_b ratio of 100) we need to supply the base with at least 240 mA/100 = 2.4 mA of current. This works out well, being that the PIC microcontroller outputs a maximum of 25 mA from each of its pins, about 10 times more than the transistor needs.

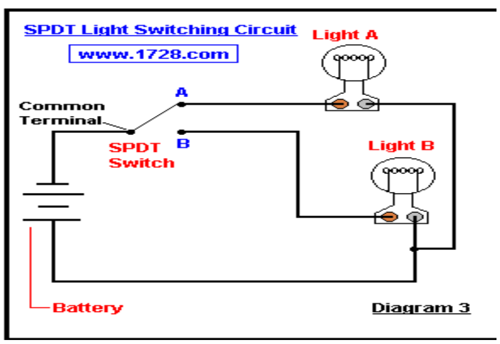
One issue with powering a relay using this scheme is that after relay is activated and shut-off, so is the transistor that is powering it. The coil of the relay acts as an inductor, and therefore follows the equation:

$$v = L\frac{di}{dt}$$

where v is the voltage across the coil, L is the coil inductance, and of course the change in current with respect to time through the inductor. Once the relay is shut off, much energy is stored in the magnetic field surrounding the coil, and has no place to dissipate; except through the transistor, possibly destroying it. The instantaneous change of current form 240 mA to 0 mA in almost no time at all drives the coil voltage very high, and it is this high voltage that damages the transistor over time. One way of getting around this is to place a diode in parallel with the coil of the relay and have the anode at Vcc, so that when the relay is shut off, all the stored energy can dissipate through the diode and relay coil and be lost as ohmic heating.

Communication Failure – The most probable failure event in regards to our design is in the event that the communication radio link which sends the "turn-on" signal to the microcontroller fails. If this were to occur, there would be no way for any of the Wi-Fi routers to power-up, therefore not connecting internet to any of the other locations. This would not only make our design not useable, but also counter-productive of what we are trying to achieve in the first place – "internet-on-demand".

One method of counteracting this is to use a switch so that we can effectively by-pass the relay altogether, and simply make the power corresponding to that relay to be active at all times. Of course in this scheme, the users would have to manually shut-down the system at times when the source of power (the sun) is no longer available.



Example of Single Pole Double Throw (SPDT) Switch

What we have devised is to use a single pole double throw (SPDT) switch to switch between the relay path or an always active path for powering the peripherals of our project. Basically this amounts to two different modes for our design: automatic for the relay path, and manual for the always active path. The above example demonstrates how such a switch operates. When light bulb A is to be activated, place the switch to position A, and vice versa for light bulb B. Think of the relay path as light bulb A, and the always active path as light bulb B.

Conclusion

For our design, we are using a radio link to transmit a "turn-on" signal to a microcontroller which will then enable location specific networking equipment to turn on so that internet is available to that specific location. In order for this to occur, relays are used to relay power from an energy source to the specific equipment needing to be turned on. Unfortunately, the microcontroller does not supply enough current to activate the relays, but this is easily remedied by using a transistor as a buffer. Sending the microcontroller current through the base of the transistor will saturate it, allowing current from the Vcc source to be used as the current to activate the relay.

In the event that the communications system breaks down (radio link), there will be a by-pass in the form of a single pole double throw (SPDT) switch to switch between the relay path and an always active path.

References

Microchip dsPIC30F1010/202X (2006). *dsPIC30F1010/202X Data Sheet.*
Retrieved April 1, 2010,
from: http://ww1.microchip.com/downloads/en/DeviceDoc/70178C.pdf

Electus Distribution (2001). *Relay Driving Basics.*
Retrieved April 1, 2010,
From: http://www.jaycar.com.au/images_uploaded/relaydrv.pdf

Tyco Electronics (2003). *S86R/S87R series Data Sheet.*
Retrieved April 1, 2010,
From: http://relays.tycoelectronics.com/datasheets/S86R_DS.pdf

Relay. In *Wikipedia*. Retrieved April 1, 2010,
From: http://en.wikipedia.org/wiki/Relay