Build Your Own
CheapNet
RS-485 Control System

This article begins a new series of projects that we believe will revolutionize your understanding about how wiring should be connected between two or more devices that are separated by more than a short distance. The device we are about to discuss is easy to use and is so versatile, that we are sure that you will easily put this project to use in many different ways. Let’s get started with some basic background info, build on it, and cover some easy to understand examples that you can put to use in the real world.

WHAT IS A NETWORK ?

A network is two or more devices which are tied together over a common set of conductors. Serial communication is the format which is used for that communication. The information can be sent Simplex (travels in one direction only) Half Duplex (travels in both directions alternately) and Full Duplex (travels in both directions simultaneously). Normally, Simplex and Half Duplex only requires a single twisted pair of wires while Full Duplex requires 2 twisted pairs of wires. The wires are required to be twisted because this allows cancellation of some of the common mode noise factor. More about that at a later time.

Some of the earliest serial communication standards in existence are still with us today. RS-232 is what most people have on their computers which lets them communicate with their mice, trackballs, and other peripherals including some printers. This standard was designed to let you communicate up to about 50 feet and you can actually go farther if the baud rate is low enough. As a matter of fact, there’s companies which sell networked RS-232 systems. But, because of the standard, they’re vunerable to a lot of problems.

First of all, the voltage level is -3 to -15 VDC for Logic 1 and +3 to +15 VDC for Logic 0. Anything between -3 and +3 VDC is considered noise. Now, since wire that co-exists in a cable are capacitative in nature, it takes a little bit of time to charge up the wire to that potential and discharge that potential on the Logic 0 level. This takes electrons to charge and discharge the cable before the reading is valid on the other devices. And that’s where the problems start.

A LITTLE BIT OF HISTORY

20 years ago, I worked as an Electrical/Electronic Specialist at Dow Chemical-Research and Development. Some of the research labs were experimenting with electronic control systems which could be programmed to change their function without changing the physical wiring itself. The device was Texas Instruments 5 TI series of Programmable Logic Controllers. The door was opened and I began to learn about the emerging field of automation and Distributed Control Systems. The following year, I began installing and programming Allen-Bradley PLC's for the U.S. Government and private industry applications.

These PLC's began to be remotely linked over high speed data networks. Allen-Bradley named their networking system as Data Highway. Another PLC manufacturer, Modicon, developed their protocol MODBUS and MODBUS +. I have worked for many engineering firms over the years before starting my own engineering business and I have personally installed hundreds of systems, so I know what I'm talking about when I speak of networks. By the way, the hardware specification of these networks is called RS-485.

THE RS-485 STANDARD

The RS-485 standard is THE way networking is done PERIOD ! Who uses networked PLC's, you might ask? All the car manufacturers, Disneyland and all other amusement parks, and just about every manufacturing plant in the world. RS-485 is a balanced transmission line which transmits at 10 Mbits/ second or 4,000 feet…but not BOTH at the same time. It normally terminates with 120 Ohm resistors at each end of the network line. The smart idea is to run at a speed that is fast enough to be practical, but slow enough to go the distance and not have problems with noise and interference !

RS-485 uses a differential balanced line with two points labeled 'A' and 'B'. The logic convention is that if A > B by .2-6 VDC then the output is logic 1. If B > A by .2 to 6 VDC then the output is logic 0. In other words, by changing the POLARITY between A & B the logic level will be high or low. The receiver is able to detect these levels as low as .2 VDC and will tolerate a differential voltage of 6 VDC. The transmitter must be capable of generating -1.5 VDC to + 1.5 VDC. That's pretty much covers the standard.

The system we have designed is a Half Duplex network which uses a 8 data bit, no parity, 1 stop bit format which runs at 2400 baud. A DIP switch sets one unit as Master and the other unit as Slave. Once the units are energized on each end, the Master will serially transmit the status of the eight inputs to the Slave which will display those bits as outputs. Then the Slave unit will serially transmit it's eight inputs back to the Master which will, in turn, display those bits as outputs. And at 2400 baud, that's happening about 240 times a second !

LIFE IN THE REAL WORLD

Farmer John (not his real name) lives in the country with his chickens, cows, and pigs. He's got a REALLY big farm and he works 7 days/week. He kinda likes to keep an eye on things from the house such as whether the door is open on the chicken coop (ya gotta watch them weasels) or whether the hen house is warm enough. Cold chickens don't lay too many eggs, ya know. And since he's got all them animals, he lives far enough away from them that he doesn't have to hear them or SMELL them !

The point is, even a low tech situation can benefit from a CheapNet. It's so inexpensive that it opens up new avenues of opportunity. You can even run the Slave on batteries and charge them from a solar cell from Radio Shack if you don't have any electricity out there. Discrete inputs (On or Off) are commonly known as switches. Magnetic switches for doors and adjustible temperature switches for the hen house.

INTERFACE DEVICES

Speaking of inputs, if you look at the other projects such as the universal alarm system, you'll see that we believe in using 78L05 voltage regulators as input conditioning circuits. It helps with noise and voltage drop problems. Input 9-24 VDC to the input of the regulator and tie the output to one of the inputs on the CheapNet module. On the output side, we suggest using the IRLZ14 HEXFET transistors. You can also use the Radio Shack reed relays (don't forget the clamping diode) like we used in the auto dialer. Or use a piezo buzzer, a LED with a series resistor, etc.

Price wise, these components are fairly inexpensive. 78L05's are 30 cents ea. in lots of 25 and the IRLZ14 HEXFET's are ~ 61 cents ea. in lots of 10: prices are from Digikey Corp. (http://www.digikey.com). You're probably not going to need all the inputs and outputs at each end. In that case, ground any unused inputs and ignore the unused outputs. If the corresponding inputs on the other end are grounded, these outputs won't be energized irregardless.

Please keep in mind the current limitations of the 78L05 which supplies power to the unit is only rated for 100 ma. You need to make sure that whatever you hook up directly to the outputs doesn't exceed this rating. Otherwise, run your load through the HEXFET transistor to a source separate from the CheapNet voltage regulator. Now let's talk a litle bit about how to configure your inputs and outputs to get the desired results.

Speaking of current, Microchip Technology recommends limiting the amount of current sourced (output returned thru gnd) to 20 ma. per pin and 40 ma. per port. They also recommend limiting the current sink (output returned thru supply) to 25 ma. per pin and 50 ma. per port. Take advantage by using the current sink option and you can control more loads per application.

Inputs can be wired as failsafe by wiring them as normally closed contacts between the power source and the input. This makes the input normally high so anything that disrupts the circuit, will cause the input to go low. On the corresponding output side, a reed relay or LED can be wired between the output and EITHER supply power or GND.

If the other side of the load is tied to supply power, the load is ENERGIZED when the output is LOW and if the other side of the load is tied to GND, the load is ENERGIZED when the output is HIGH. For example, if you wanted a failsafe Input/Output configuration using Input 1, you would wire a Normally Closed set of contacts between the supply power and the 78L05 regulator feeding Input 1. An LED and current limiting resistor are wired in series between Output 1 and supply power. Now, when the contacts open, the LED lights.

You can have the same failsafe input and have reverse action on the output by wiring Output 1 to GND instead of supply power. You can also tie inputs together so that one input can control several outputs. For example, you can wire Input 1 and Input 2 together, use only one 78L05 and have Output 1 ENERGIZED when the input is HIGH and have Output 2 ENERGIZED when the input is LOW by wiring the outputs as mentioned above. And you can take the outputs thru 1N914 steering diodes to energize a device whenever 1 of several outputs is either HIGH or LOW.

This system allows you two different ways to monitor that you are communicating properly with the other end. The first way is built in thru the Watch Dog Timer. When the PIC is programmed there are several configuration options that you can select from. One of them is the Watch Dog Timer or WDT, for short. The WDT is a free-running oscillator that will RESET THE PROGRAM IF IT TIMES OUT (~ 2.3 sec.) What we have done is to use the CLRWDT instruction at the end of the READ SERIAL DATA portion of the program. This resets the WDT constantly about every 10-11 msec. So, if the nodes aren't communicating, EACH node will reset itself until it reinitiates valid communication. Part of that RESET is to clear the outputs; therefore, turning everything OFF when the nodes quit talking. This allows us to monitor and flag a COMM FAIL on the network.

The second way to monitor involves either tying an input to an output on the same node to verify the output is turned on, or monitoring a switch status that responds physically to the output command. For example, a spare set of contacts on a relay could indicate that the relay physically operated when it was energized. Or, let's take farmer John's example above, when you sent the command to heat up the hen house, the temperature switch would open (asuming failsafe wiring) when the temperature increased to setpoint.

I won't bore you with all the possibilities available to you because there's so many different sensor interface circuits. Most are built on Op Amps with discrete output levels(On or Off) to monitor greater than/lesser than parameters of temp, pressure, liquid level, light level, motion, flow, windspeed, humidity, position…I think you get the picture. This lends itself well to remote monitoring of security systems, smoke detectors, etc. and will even interface quite well to our Universal Alarm System, Audio Annunciator, and Autodialer projects. We design many of our projects as interconnecting 'Building Blocks' to allow YOU to achieve the greatest amount of creativity !

HOOKING IT UP

Now, about the wiring. I strongly suggest using a # 24 AWG twisted shielded pair wire as the network cabling. This is pretty much what is used as the standard. I won't tell you that other wiring won't work-because sometimes it might. You've got to determine that on your own based on environment and distance. But why take chances and ask for trouble ! Since the wiring should be shielded, the shield needs to be GROUNDED to an earth ground, such as you have in a standard 3 wire 120 VAC supply. This is the GREEN wire, not the black (hot) or white (neutral). The shield is to be grounded on one end ONLY and isolated on the other end. Typically, it is grounded at the Master node.

STARTUP

A little common sense here will save you a lot of headache. Don't get in such a hurry that you forget a few of the fundamental rules. You're installing a network system which needs to be started up in a specific sequence. First of all, make sure that your power supply is well regulated and has a fast rise time when it is energized. The MCLR reset line is tied to the supply voltage. If it takes too much time for the power supply to 'turn on' the chip may not reset properly on startup. I STRONGLY SUGGEST that you put an ON-OFF switch between the power supply and the chip AND install 0.1 uf. capacitors accross the chips to filter out 'noise spikes'. This will take care of the problem.

Second, the CheapNet controller chip is a CMOS device. DON'T energize ANY of the inputs until the the supply power is turned on AT EACH END. Perhaps you can put another ON-OFF switch to control power to the input devices on each end. And, of course, make sure the comm wiring is in place between the units. Turn the DIP switch ON for Master and OFF for Slave mode. It really doesn't matter which end is which.

We are offering a complete kit containing 2 ea. preprogrammed PIC 16C55 CheapNet IC's, 2 ea. Panasonic 4 Mhz. ceramic resonators, 2 ea. 75176 RS-485 tranceiver IC's, 2 ea. 78L05 voltage regulators, and 2 ea. 120 Ohm resistors @ $ 30.00 including shipping & handling within the continental United States. We sell all our products worldwide. Please contact us for details.