16 Channel PIC Microprocessor Based Computer Controlled Christmas Light Dimming Controller with SSR's.

Disclaimer

(http://forums.parallax.com/forums/default.aspx?f=21&m=61194)
This circuit can ONLY be used for dimming lights. Inductive loads cannot be used. This includes electric motors and fluorescent lamp ballasts. The reason is that the reverse EMF produced by inductive loads produces an out-of-phase wave back to the triac so that the triac never sees zero volts. Thus the triac never ceases conducting, resulting in total loss of control. There are snubber circuits that can be added to filter this reverse EMF and restore order to the triac.

As with all circuits that use 120 volt AC mains, the circuit should properly fused for the loads to be used. Due care must be taken when working AC Mains voltage to protect from shock and injury. Never work on the circuit with Mains voltage applied to the circuit. Do not bring Mains voltage onto a breadboard. It is too easy to forget where on the breadboard these voltages are.

You can put the AC side of the circuit on a perfboard or prototype board and bring the logic level signals to/from a breadboard. Just remember to use adequately sized wire in the AC side for the anticipated current.

There was no formal analysis of this design done in either the analog or digital domain. It is not guaranteed to work for the expressed purpose. No warranty is expressed or implied. This was a hobby project that I simply wished to share, at cost, with others at no profit to myself. In short, I take no responsibility for personal injury, injury to property or any other damage or injury related to the prototyping or use of this circuit. Proceed at your own informed risk. Electricity can kill you. This circuit can burn down your house and then kill you, or maybe in the other order too!

Please keep in mind that this circuit was designed with 120VAC 60Hz in mind. Other voltages and frequencies were not considered. However, if you do try to explode, I mean use, one of these boards under strange and new conditions; I would be interested in knowing the result of the experiment. (Please include projectile velocity in your report).

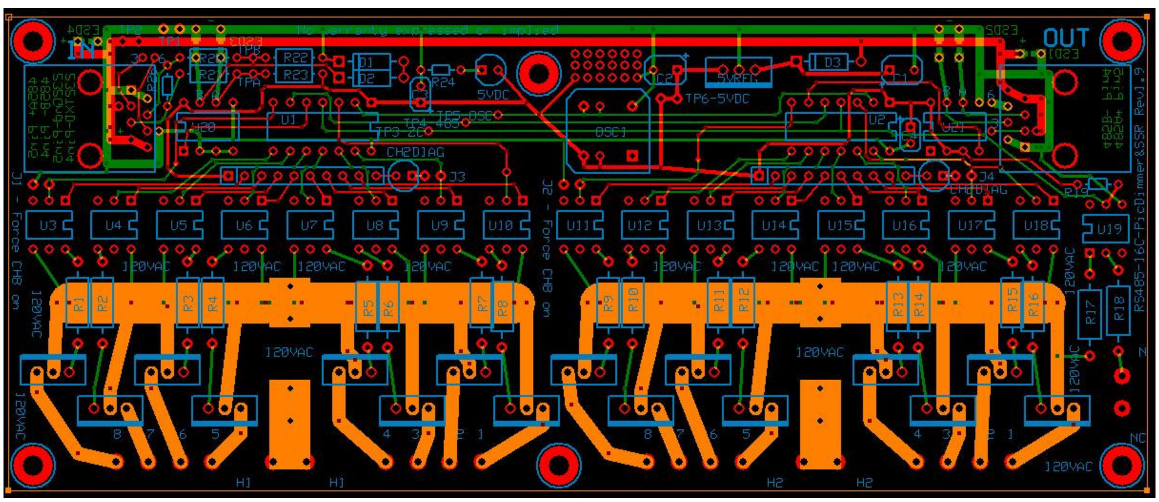
Every effort has been made to select good components, but as with everything else, no warranty is expressed or implied.

Introduction

First and foremost I want to make it perfectly clear that I am not the expert on these topics. Dozens of people have contributed to the creation of this document; first and foremost **Phil Short** should be attributed with most of the wisdom! Secondly **Sean Bowf** and **tfmacz** from picdimmer.17.forumer.com provided a great deal of patient technical support. I am simply a messenger who would like to see a clear document that contains the information all in one place to save poor Phil (And now myself) the stress of answering the same questions 100's of times! I also believe altruistically that this contribution will help push the technology to higher limits. By building a cook-book for these steps, I hope someone else will invest hundreds of hours and use this design as a starting point, to take it to the next level!

If you find errors in this document email: jaunemaillot@gmail.com

A picture of the 7" x 3" Compact Layout



The 16 Channel Renard Pic Dimmed SSR Design

This picture is intentionally small, so I don't have to update it constantly as the design changes. A PDF of this layout (updated as necessary) can be found here for detailed review. The PDF is the most viewable version, you can zoom in nicely.

http://www.ritzfam.com/ChristmasMadness/Renardx16SSR/Release

Why SSRs on board? Why 16 Channels?

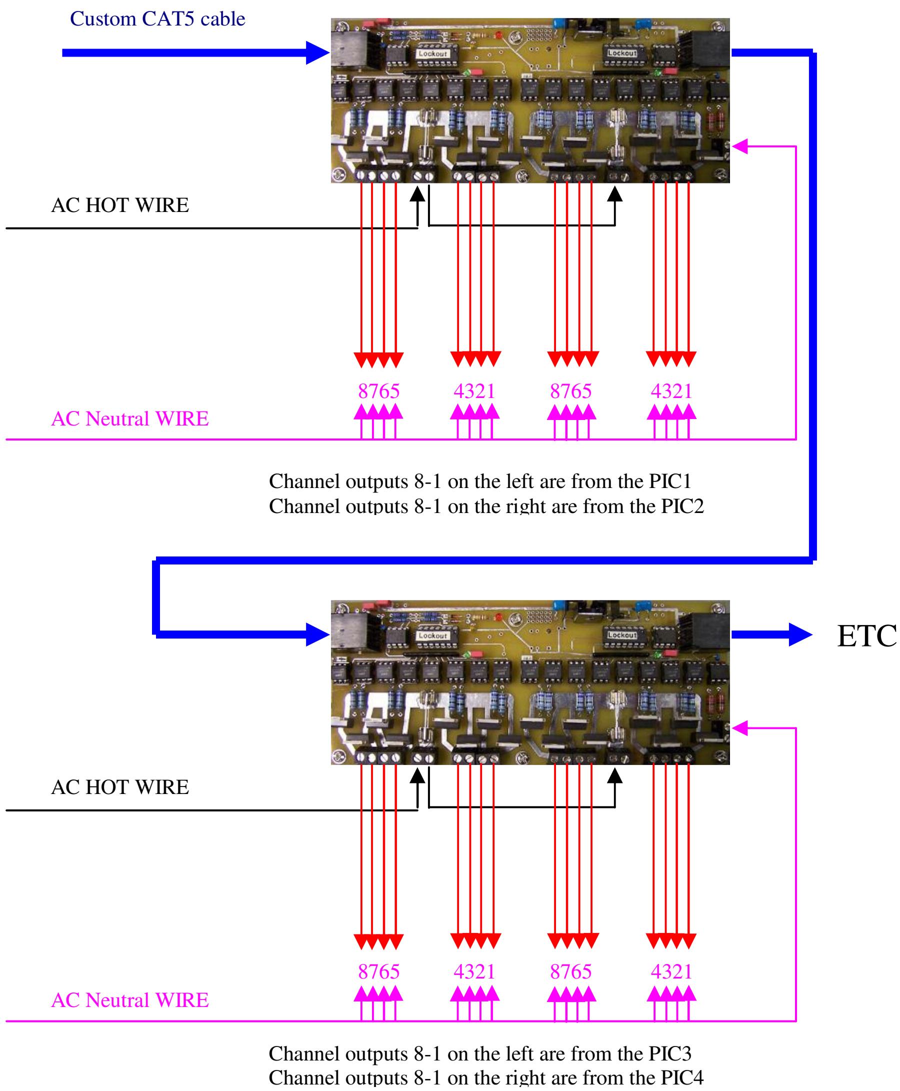
In my case, all of my lights will be in neatly spaced discrete groups. I will have 16 mini-trees located very close together. I will have 16 "greeting candles" in the front window (Small single bulbs that look like candles), and the lights on the front of my house will break up nicely into two clean groups of 16. In addition, 16 channels are perfect for the lights on the roof. I like the idea of a 16 channel SSR onboard because essentially I will only need 5 AC extension cords. One cord will run from the power mains to each enclosure and I wont need to build 36 SSR boxes for outside use. In addition I will purchase SPT-1 from actionlighting.com (500 feet of 16 gauge wire for $55) to run from the screw down connectors on this SSR board out to the lights, using snap on plugs designed for outdoor use. This in my opinion is a very low cost, clean solution to my lighting criteria. Having said that: this may not be the right solution for everyone. Certainly one downside of this approach is that you may not be able to upgrade your controller from year-to-year as new improvements are made. What I mean by that is that since you have invested $60 in the controller (with SSR's) you are "stuck" with this design. Naturally the board is very flexible and a lot of changes can be made, and the pic firmware can be changed as well… But I'm just providing you with "full disclosure".

Two things that I view as a disadvantage of the "controller separate from the SSR" designs are as follows:
1) You need to run AC power to all of the SSR boxes. That is a lot of green extension cords! There are several interesting designs to handle this (The most recent being the strip board being discussed on computerchristmas.com). I do understand that this is the route many folks have gone, and it is not a bad method, and depending of the layout of your yard it may even be better. I'm only suggesting in my case this design is more efficient. I also don't like the idea of putting low votage circuitry "under" live 120 volt receptacles, and worrying about water getting into SSR boxes outside.

2) You need to run CAT5 to each SSR from the controller. For a 64 channel design, with the SSR's averaging 50 feet from the controller, that is 800 feet of Cat 5! With this design, you chain the controllers serially. Lets assume 100 foot spacing of the controllers, you would need 400 feet of Cat 5. But more importantly you only need to make 4 cables instead of 16.

The ramifications of this design are: Cheaper deployment of lights out to farther distances from the controller. RS485 is good up to a kilometer, and you would only need a single cat 5 cable to run that distance… I just want to restate, Olsen 595, and Renard controllers are not a bad design, they work great. But in my case, this design better meets my needs. The Olsen and Renard designs with external SSR's would be ideal when the distance from the controller to the SSR's is not too far, or if the spacing between lights was very large.

The Big Picture – How the to hook it all up



Cost to Build

The cost of the PCB itself will be between around $20 each, perhaps as low as $18 if enough people join in. I will make no profit, I will sell them at cost, plus shipping, and paypal fee's. The parts are going to be $46 per board (not including shipping and tax if applicable) from Mouser. However, $13 of the $46 is for optional components. You can decide what to install if you would like to keep your costs down. Having said that, sockets for the DIP's and LED's really should not be left out of the design, you could cause a lot of pain for yourself by leaving those out. Read the comments in the parts list, I've indicated which optional parts are truly optional.

The parts list can be found here:
http://www.ritzfam.com/ChristmasMadness/Renardx16SSR/Release

Look for the latest revision of this document:
16cPicDimmedSSR_Parts_List_RevX.X.xls

How much power can it handle?

There are online trace power handling calculators, but they don't agree, and neither can we. The traces are on the top and bottom of the board as well as being as wide as the design will allow. The power distribution traces are ¼ inch wide on the top and bottom. The traces out to individual triacs, and out to the screw down connectors are .08 top and bottom. ExpressPCB claims to have a 1.25 oz/ft^2 copper process. The triacs appear (From the experiments of those who do this regularly) good for up to 1A without a heat sink. And It seems reasonable that you might be able to get up to 3 or 4 A (each channel) with a good heat sink. I think someone (me?) will need to sit down and run a duration experiment until one of these things explodes, only then will we know the true maximum limit. For now, the parts list will ship with a 5A fuse (5mm x 20mm common glass tube type), and you can make your own decision about how to handle the power issue.

A template to create your heat sink is provided here:
http://www.ritzfam.com/ChristmasMadness/Renardx16SSR/Release

Look for the latest revision of this document:
16CPicDimmerSSR_HEATSINK_TEMPLATE_300DPI_RevX.X.jpg

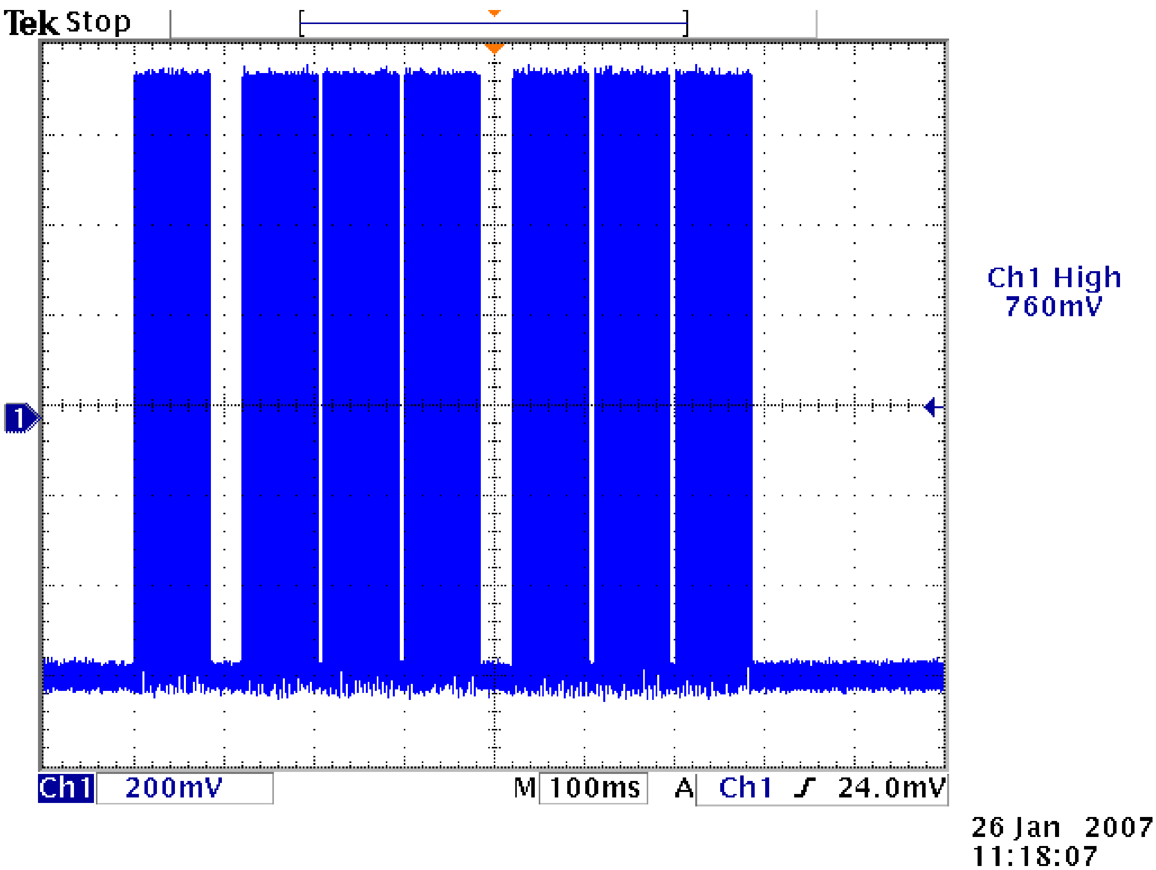
How much power does it use?

For this configuration (May not be identical to yours)

5VDC Power Requirements	
	mA
LM7805	8
ST485EBN	20
MOC3023 - 16 * .008 / 256	0.5
H11AA1 - 5V / 27K	0.2
PIC16F688 =(HFINTOSC mode = 8 MHz)	10
Osc	35
Total with option Osc	73.7
Total without Osc	38.7

The traces on the board are setup such that pin 3 and 6 on the RJ45 connector are spares. Should you require them for more current carrying capacity to the controllers, there are traces close to the related RJ45 connector pins that you can just drop a ball of solder on, and you can now have 3 VCC wires and 3 GND wires in your Cat5 cable. Look at the schematic at pins 3 and 6 on the input and output RJ45 connectors.

How many channels can I use at once?



Phil Short and I have taken this design to 38400 bps with no problems. Thus at 25mS resolution you can drive 64 channels through a single serial port.

The math:
Theoretically, at 19200 baud it is possible to get up to 192 channels using 100ms resolution, but in the real world, with 100ms resolution 128 channels use 85-90ms of the 100ms cycle time. Thus you are at a 85% duty cycle when running 128 channels. In the picture above, Vixen is driving the serial port (RS232) with 100ms resolution and 128 channels. The width of the squares is 100ms, you can see what I mean by the statements above. In order to go beyond 128 channels at 19200 baud, you have a couple of options. The fastest and easiest is to just use another serial port. For each additional serial port you add, you can use another 128 channels. Vixen can natively support up to 4 serial ports, and some computers already have two serial ports on the back. Another option is to reduce your resolution from 100ms to 120ms for example. That would give you room for (theoretically) 20% more channels.

Phil and I continue to investigate using the external oscillator to push the frequency higher than 38400 bps, but that is a little bit harder, and will take some time and energy. I think we can have something running 57.6Kbps by April perhaps, maybe faster.

8 channel Option

You could setup this board for use with anything from 1 to 16 channels. The simplest modification is to use it for only 8 channels. In the case of using only 8 channels, simply do not load: The second pic, the 8 opto-isolators, and 8 traics, and 8 triac resistors, and the SIP resistor for the opto's. Short pin 5 and 6 where the PIC would go so the RS485 in and out are connected together and now you have a 8 channel Renard with SSRs on board.

Obviously your cost per channel will increase with this option, however, this is a fairly elegant solution, and I will be making several of these for myself.

Test Points and Jumpers

This section describes the function and use of test points and jumpers in this design.

3, 6 These are the spare RJ45 wires. These really aren't true test points. Can be easily soldered to connet to nearby vias to increase the number of wires in the CAT5 cable used for power distribution to the controllers.
J1, J2 You may optionally solder a two pin header here, and install a jumper in order to force pin2 of the opto-isolator to zero, thus turning on the triac for channel 8. This is a good thing to do to prove that 120 volts is active, and that everything downstream of the opto-isolators is working correctly.
J3,J4 You may optionally solder a two pin header here, and install a jumper in order to enable the LED (optional) for use with the PIC diagnostic code. You probably will not want to use your controller continuously with this jumper installed as it will likely draw down the voltage a bit to the opto-triac and may cause channel 2 to turn on/off unpredictably when compared to other channels that do not have the LED. Also, it might cause channel 2 to appear at a different intensity. (Note, it seems to look fine installed for me, but you never know, depending on your load and conditions, it might cause a problem, I personally am running my controllers with this jumper installed)
TP1 Ground - You may optionally solder a two pin header here to attach a ground for a oscilloscope or voltage measurement etc.
TP2 VCC - You may optionally solder a two pin header here to measure the voltage to the controller via the RJ45 lines. Remember that this voltage is between 9 and 12 VCC and is a supply to the 5 volt regulator onboard the controller. You may cut the trace between the two pin header, to directly measure current, but if you do this, you will require a two pin header and jumper in order to "repair" the board when you are finished.
TP3 Zero Cross - You may optionally solder a single pin header to this point in order to look at the ZC signal on an oscilloscope.
TP4 UART. - You may optionally solder a single pin header to this pin in order to examine the internal serial signal between pics with an oscilloscope.
TP5 Oscillator – You may optionally solder a single pin header to this pin in order to examine the output of the oscillator which goes to both PIC's.

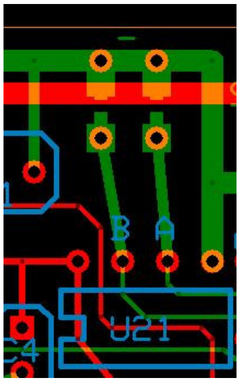
TP6 5V Reg – You may optionally solder a two pin header to this pin in order to measure the voltage out of the 5 volt regulator. You may cut the trace between the two pin header, to directly measure current, but if you do this, you will require a two pin header and jumper in order to "repair" the board when you are finished.
TPA RS485 A – You may optionally solder a single pin header to have a test point to view the RS485A signal with an oscilloscope.
TPB RS485 B You may optionally solder a single pin header to have a test point to view the RS485B signal with an oscilloscope.

You will notice an array of 18 round hole VIA's near the 5VDC power on indicator LED (Top center of the PCB). 5VDC is connected to the bottom left via, and ground is attached to the bottom right via. You can use these for anything you like.

You will notice a few extra via's in the oscillator footprint. It is possible that the oscillator my never be implemented. I have added a few more terminals for ground and power so this area could also be used for prototyping. This was done because, if the oscillator is never used, this free's up pin two on both PIC's. You could cut the trace between the two VIA's under the oscillator footprint, and use the two pin 2 connections independently in conjunction with the prototype area.

Reminder, there are also two spare unused wires in the RJ45 cable. Look for the two via's near each RJ45 connector labeled "3" and "6". There is additional space and features in this board to allow you to develop and prototype within the PCB.

ESD



The parts list assumes you will install RS485 chips that have electro static discharge protection (ESD). However, if you choose RS485 parts that do not have ESD protection, or you wish to install your own ESD devices for any other reason, this section will explain how to do so.

ESD2, ESD3
One option in this design is to install surface mount electro static discharge protection to protect the interfaces called “PESD” (Polymer ESD). On the bottom side of the board, near the A/B pins of both the input and output RS485, and underneath the RJ45 connectors, there are surface mount regions labeled ESD1 through ESD4.

Please note the “+” or “-“ which indicate the VCC and VSS sides of the region respectively. Some ESD components may have a + and – side (polarized). In each RS485 ESD region you may choose to install an ESD device for RS485-A and ESD device for RS485-B. The A and B signals should go through one ESD device to GND. In the picture above, the surface mount components would be mounted vertically.

ESD1, ESD4
Even if you use RS485 parts that have ESD protection features, the 12VDC power rail will still be unprotected. There are ESD pads below the RJ45 connectors. They can optionally be loaded with a MOV (Metal Oxide Varistor), between the 12VDC rail, and the GND rail. This will protect the 12V power rail from ESD.

It should be noted that the sole purpose of this ESD protection is to protect the interfaces of the controller, that is, the RJ45 connections only. These ESD countermeasures assume that the ESD pulse is originating from the RJ45 connector. If you have a static charge on your finger, and touch the controller, you may still cause damage. Every time you open up your controller box, and you need to touch something (First make sure the AC power is disconnected) you should momentarily touch your finger to TP1, the test pin that goes to ground to discharge yourself.

How to hook up AC power

At the bottom of the board you will see 22 terminals as follows:
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
8 7 6 5 H1 H1 4 3 2 1 8 7 6 5 H2 H2 4 3 2 1 (around corner) “unlabeled” N

It is assumed that the “white wire” or neutral wire in the AC system will never be attached to the printed circuit board but will all be attached together in a wire nut (not included in parts list) or a wiring block. The only exception to this is that a single neutral wire must be connected to the “N” terminal in order for the zero cross circuit to work. The reason there is an extra terminal next to the N terminal is because having a single terminal is bad for mechanical stability. Having two soldered to the board makes it sturdier. The unlabeled terminal is not connected to anything. I could have connected it to neutral as well, but I had visions of someone screwing a black wire in there, assuming that is where the power goes in… POOF.

When attaching the “black wire” or hot AC wire, in the simplest case, it would be connect to one of the H1 terminals, and then another (preferably black) wire would go from the other H1 terminal to one of the H2 terminals. In this way, you are supplying 120VAC to the “center” of both 8 channel groups.

All of the “hot wires” for your lights (the outputs of the SSR’s) are screwed down on terminals 8-1 on the left, and 8-1 on the right. These are your “channels”.

If it turns out that each side of this controller does not explode when you run 15 or 20 amps through it, (assuming you have installed good heat sinks), then you could potentially have two separate feeds from your circuit breaker box (Must be GFI protected for outside use) and provide a separate 15 or 20 Amp feed to the H1, and a separate 15 or 20Amp feed to H2. Having said that, I have no idea under what conditions this board will become a flaming projectile. Experiment at your own risk. I personally error on the side of safety and will only be running about 1/3A to 1/2A per channel, but it was fun to completely over design the board, and it will be novel and interesting to find out at what amperage the board ignites. Also, it might be important to have both feeds to the controller be on the same leg of your 220, so they are in phase with each other. But I’m not sure. If you have no idea what I’m talking about, then this particular modification isn’t for you. Don’t try this at home folks.

The Renard Project

Renard Who?

(http://picdimmer.17.forumer.com/viewtopic.php?t=3)

Phil Short developed a mini-project involving a PIC-based Computer-Controlled Light Dimmer, mostly for controlling incandescent display lights (such as Christmas displays). This light dimmer is a low cost, low chip-count solution using a commonly available PIC microcontroller to control (on, off, and dimming) up to eight lamps (or light strings) per controller. Phil named the project "Renard". In his words "Vixen -> fox -> renard".

The basic idea is that someone can use the serial port on a PC to control and animate a light display, often synchronized to music. For example, a light display using exterior Christmas lights could show Santa and a sleigh with reindeer moving across a lawn or the face of a house in time to 'Rudolph, the Red-Nosed Reindeer' or some other relevant Christmas song. There are many other possible uses as well, including other holiday or more permanent displays.

This project involves only one piece of the puzzle - the low-power control electronics that accept commands from a PC and provides on/off/dimming signals. It is a DIY (Do-It-Yourself) solution, since there are not currently any kits or finished products available. More information about this sort of process can be found on the web, including the http://www.computerchristmas.com site. The basic PIC controller can control eight lamps, but multiple PICs can be daisy chained together to control up to 128 or more lamps through a single serial port on a PC.

Some other parts that are needed can also be found of the web, including PC software for choreographing and running the show (http://www.vixenlights.com, and solid-state relays (SSRs) (http://computerchristmas.com) for supplying power to the lights.

Since this is a DIY design, the user can select various communications methods to suit the interface and distance requirements, including RS-232, RS-485 and current loop.

Here is a link to Phil Shorts how-to at computerchristmas.com which describes how to build a 8 channel controller which uses external SSR's.
http://computerchristmas.com/?link=how_to&HowToId=71

Firmware

One of the things that distinguishes this approach at controlling lights from others, is that this project uses a PIC controller, and therefore requires firmware. The most stable link to the current firmware (and diagnostic firmware) is
http://computerchristmas.com/?link=how_to&HowToId=71

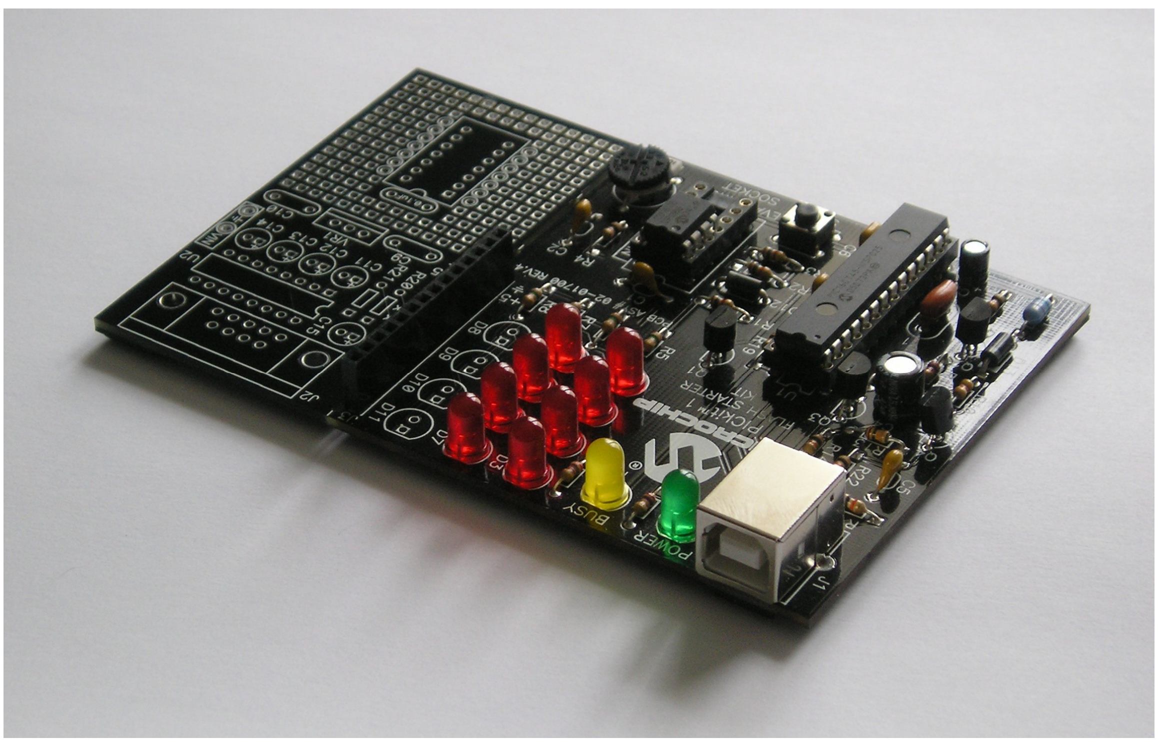
I won't try to reproduce Phil's how-to here, but I add that the diagnostic code flashes the "heartbeat" LED every second. This is done on channel 2, PIC pin 13.

In this PCB design, both PIC sockets can be used as diagnostic sockets if you apply Jumper J3 or J4 (depending on which PIC you are testing). If you drop your PIC programmed with the diagnostic code into either socket, there is a LED hooked up to pin 13, and it should blink (and turn lights on and off on channel 2 if AC power is available).

Which PIC?

The short answer is PIC16F688 mouser part number 579-PIC16F688-I/P. The long answer is that, while other versions of the PIC16F699 exist, this part number is sufficient to run within the temperature ranges we expect to see.

To get the firmware onto the PIC:



(If you are familiar with other pic programmers, feel free to email me a few paragraphs, and I can put it in here, as long as it is directly related to the Renard project)

It should be noted that specifically with the PICkit1, you will require both the "PICkit1 flash starter kit" software to program the PIC, and also MPLAB IDE in order to compile the assembly language provided by Phil. I have tried to program the PIC using MPLAB (It says it supports it) and wasted a lot of time, and it never actually worked.

1. Get a pic programming kit (These instructions assume you are using a PICKit 1)
2. Install MPLAB IDE
3. Configure – select device – choose the PIC16F688
4. Programmer – select programmer – PICKit1
5. Programmer – connect to programmer
6. Download the latest Renard code and open it in the MPLAB IDE
7. "build" to create the hex file
8. Using the PICKit 1 program (Named "PicKit 1 Flash Starter Kit"), import the hex
9. "write" to program the PIC, and then "verify" to check that it was a success.
10.

The simplest way to verify that you have programmed the pic correctly is to start with the Diagnostic code, program your PIC. Apply power and ground to Pin 1, Pin 14 respectively. Then attach a LED in series with a resistor (~500Ohms) to pin 13. This is the heartbeat output and the LED should blink on and then off every second.

PIC Output Pins and Channel Numbering

This was taken straight from the comments in the Renard firmware:

PIC PIN # Channel Output

```
;;  PIN 3  (RA4) - triac driver 0 (output)
;;  PIN 13 (RA0) - triac driver 1 (output)
;;  PIN 12 (RA1) - triac driver 2 (output)
;;  PIN 11 (RA2) - triac driver 3 (output)
;;  PIN 10 (RC0) - triac driver 4 (output)
;;  PIN 9  (RC1) - triac driver 5 (output)
;;  PIN 8  (RC2) - triac driver 6 (output)
;;  PIN 7  (RC3) - triac driver 7 (output)
```

In the 16 Channel Renard with SSR design, the channels are numbered 1-8 which correlates to triac driver 0-7 respectively. You will notice there are two banks numbered 1-8 and 1-8 rather than 1-16. This is because each pic has no knowledge of other pics and therefore every pic thinks it is controller channel 1-8. Please see Phil Shorts original how-to to better understand why this is the case.

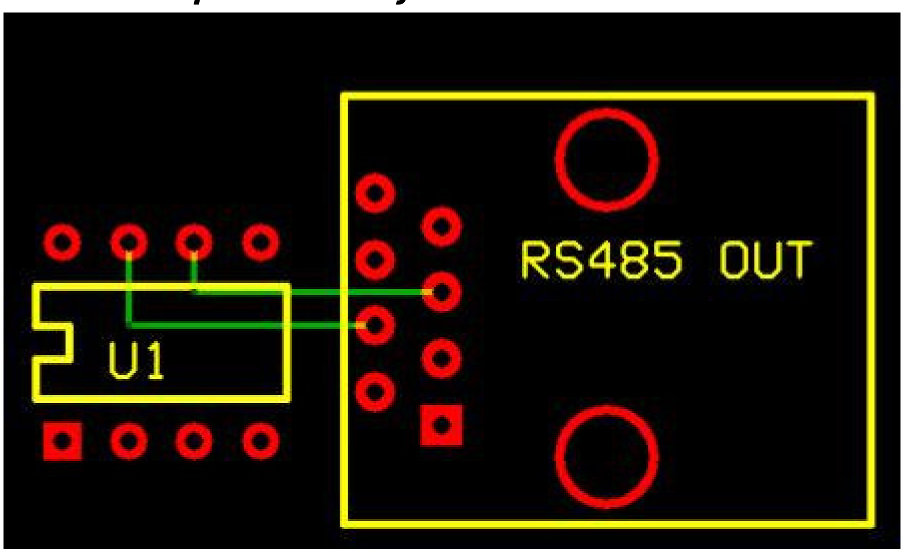
PIC Input Pins

Again, straight from the comments in the Renard firmware:

;; PIN 2 (RA5) - reserved (maybe for clock input)
In the 16 channel Renard with SSR design I have made provisions for 1/2 size crystal clock oscillator. It may be possible that Phil will release new Firmware to support higher data rates via the serial connection. This part is optional but may be a nice feature to have in the future.

;; PIN 4 (RA3) - zero-crossing (input only)
Please see the section below related to Zero Crossing.
;; PIN 5 (RC5/RX) - uart_in (input)
Please see section below related to RS485 operation.
;; PIN 6 (RC4/TX) - uart_out (output)
Please see section below related to RS485 operation.

RS485 Output Circuitry



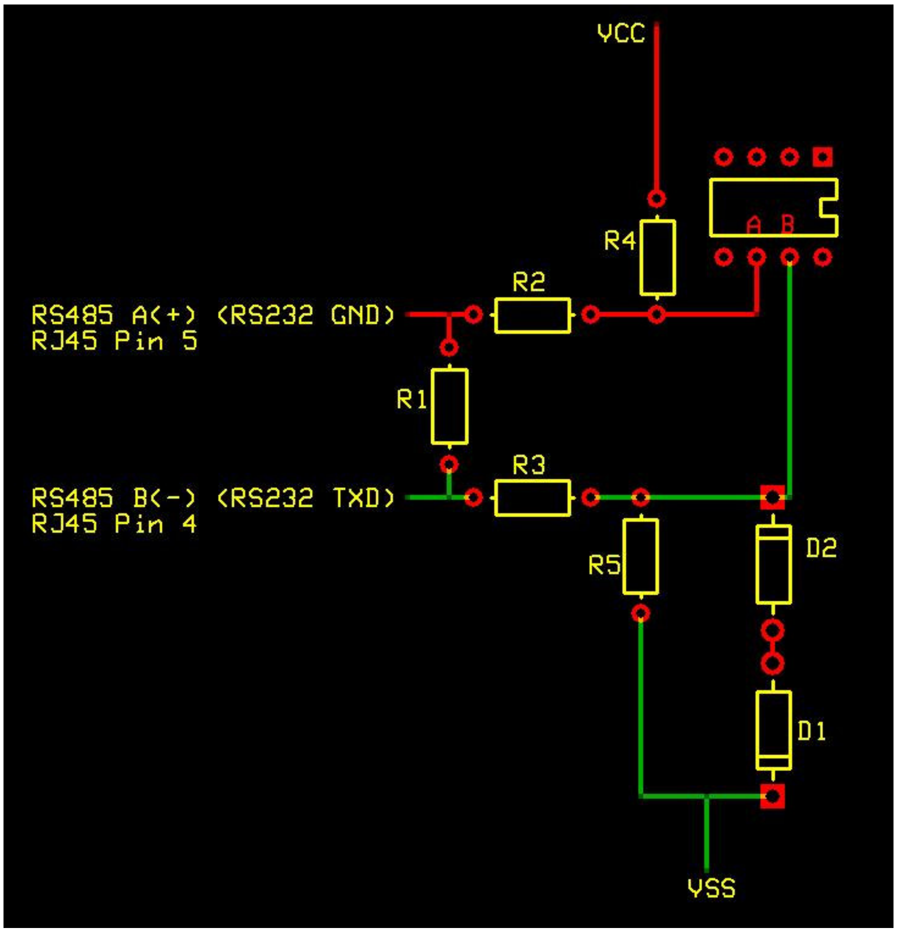
U1 = RS485 Chip

The RS485 chip in this specific case is the ST485EBN because Phil favors the low power consumption, but others are pin compatible and slightly cheaper.
Compared with the RS485 input protection circuitry, this is completely trivial.
Just get the polarity right.
RJ45 Pin 5 goes to the RS485 chip pin 6, the A or + terminal
RJ45 Pin 4 goes to the RS485 chip pin 7, the B or – terminal.

Once you go through one of these controllers, there is no longer any discussion about RS232. If you used RS232 into this board, it is now RS485 coming out. End of story.

RS485 Input Circuitry

In my opinion this is one of the hardest circuits to understand in the entire controller. I hope to alleviate some of this confusion.



R1 = 120
R2, R3 = 1K
R4, R5 = 27K
D1 = 1N5229
D2 = 1N5239
U1 = The chip is any RS485 chip.

You can run this board in two separate serial configurations. The first and simplest is to run RS232 from your computer into the first controller. If you do this, you wont be able to put your controller too far away from your computer. I would stay under 20 feet personally. You could also go out and buy a RS232 to RS485 converter ($3 on ebay+$3 shipping) for each serial port you plan on using. In this case you could have up to a kilometer (4000 feet) between your computer and the first controller. The controllers all transmit RS485, so the maximum spacing between controllers is always 1KM (4000feet) regardless of if you start with RS232 near the PC.

The front-end circuit was inspired by the TI document SLLA036B, entitled "Interface Circuits for TIA/EIA-485 (RS-485)".
http://focus.ti.com/lit/an/slla036b/slla036b.pdf

In order to make the board input universal, discrete components are being used to allow the RS485 receiver to directly receive RS232. RS232 uses alternating plus and minus voltages to signal mark and space. So depending on the computer serial port you could have anything from +/-3 volts to +/-15 volts on the wire. The RS485 uses differential "0" and 5 volts to signal mark and space. so directly connecting the RS232 transmitter to an RS485 receiver is deadly to the receiver. It will choke or croak on the -5 - 15 volts.

The main purpose of the resistors (two 1K and two 27K) is to ensure that the output of the RS495 chip goes to a known and reliable state when the input cable is either open or shorted. The A pin is pulled-up, and the B pin is pulled down. While the two 1K resistors limit current in the event of a direct short.

The purpose of the Zener diodes is to prevent the input of the 75176 from going outside of the allowed range when the circuit is driven with an RS232 signal.

The 120 Ohm resistor is for matching impedance as 120 Ohms is the typical impedance of twisted pair. The 120 Ohm resistor is placed between pins 4 and 5 of the input RJ45 connector.

If you are using RS485 for PC to controller then the only thing you need is the 120 ohm resister between the A/B leads on the 485. If you are using RS232 then the best choice is to put an RS232 chip at the input. MAX232 or any of the many others. But in “Phil’s design” (The universal design) he does a couple of things to make both possible. The 27K resistor to connecting the "A" to VCC and the "B" to ground are what they consider "weak Pull-ups". They are there to ensure the input is in a known state if the computer is disconnected. That is their only function.

The 1K resistors are there to limit the current drawn from an RS232 transmitter if it is connected. (They also limit current if an RS485 transmitter is connected but don't hurt the situation).

If an RS232 transmitter is used the "TXD" lead is connected to the RS485 "B" lead. The "A" lead is pulled to VCC(+5V). The output of the receiver switches when the "B" input goes above and below 5 volts. Hopefully by looking at the two pictures above (RS232 scope trace, and RS485 scope trace) it will be obvious why this is the case.

The Zener diodes from the "B" lead to ground are there to limit how far negative the "B" lead is pulled when an RS232 transmitter is connected.

The data sheet on the 485 lists "Receiver Input Voltage ... -8V to +12.5V" These are the "Absolute Maximum" values. In reality don't go there.

A Zener diode voltage drop is around .7V forward and 5V reverse (assuming a 5V Zener). so putting the diodes back to back is a way to limit the voltage in both polarities.

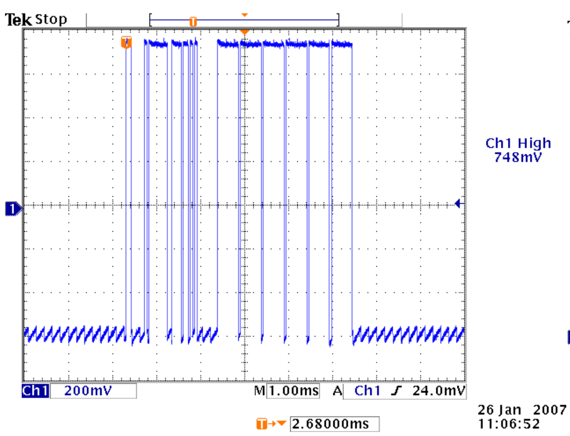
The 1N5239 is a 9.1V Zener and the 1N5229 is a 4.3V Zener so the net effect is to limit the "B" pin to a maximum of +9.1volts and -4.3volts. 1N5229 (4.3 V Zener, limits 485 input to -5.0 V), 1N5239 (9.1V Zener, limits 495 input to +9.8V)

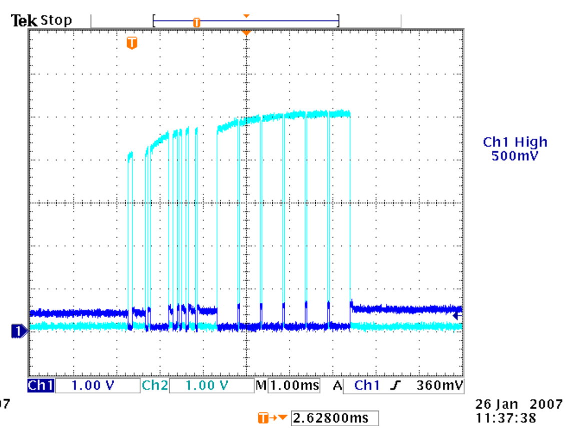
The 1K resister in series between connector and the "B" pin limits the Zener current just like in any Zener regulator circuit.

In summary:

Don’t mix up pins 4 and 5 on the RS485. Look at the pictures above closely, read, understand, and play by the rules.

RS232 VS RS485





Above are two pictures of an identical 8 channel Renard output from Vixen. The left image is the RS232 protocol, the right is after it has been sent through a RS485 converter.

We will start with a short discussion on RS232. In RS232, there is ground, and TXD (Transmit data, the dark blue line in the picture). TXD goes from ground to VCC. Ground is ground, and is therefore intuitive and pretty simple. In the picture above you can see the divisions are 200mV so in this case TXD goes from 0 to 1.5 Volts.

In RS485, the A signal (dark blue) goes from 0VDC to +.5VDC, while the B signal (light blue) goes from 5VDC to 0VDC. The resting or “no data” state of RS485 is A = .5VDC and B = 0VDC. The active state or “data being sent” state of RS485 is when A= 0VDC and B = 5VDC. Thus you can see, the signals are differential, one goes high, and the other low, for the transfer of one bit of data.

Triac

(http://forums.parallax.com/forums/default.aspx?f=21&m=61194)

The triac is the device that will switch the AC. The terminals on a triac are labeled Main Terminal 1 (MT1), Main Terminal 2 (MT2) and the Gate (G). From a black box perspective, the triac has four important functional characteristics (assuming the device is operating within its maximum ratings):

1. The triac will conduct in either direction of current flow. This means that current can flow from MT1 to MT2 or from MT2 to MT1.

2. The triac will only begin conducting when triggered by a current applied to the Gate.

3. The triac automatically ceases conducting when (only if) the voltage across and current through MT1 and MT2 goes to zero, after which the triac must be re-triggered to begin conducting again.

4. Once triggered, the triac will continue conducting until the next zero voltage crossing point in the AC sine wave regardless of the voltage/current applied to the gate. So, once triggered, the Gate input is in a "don't care" state until after the next zero crossing point.

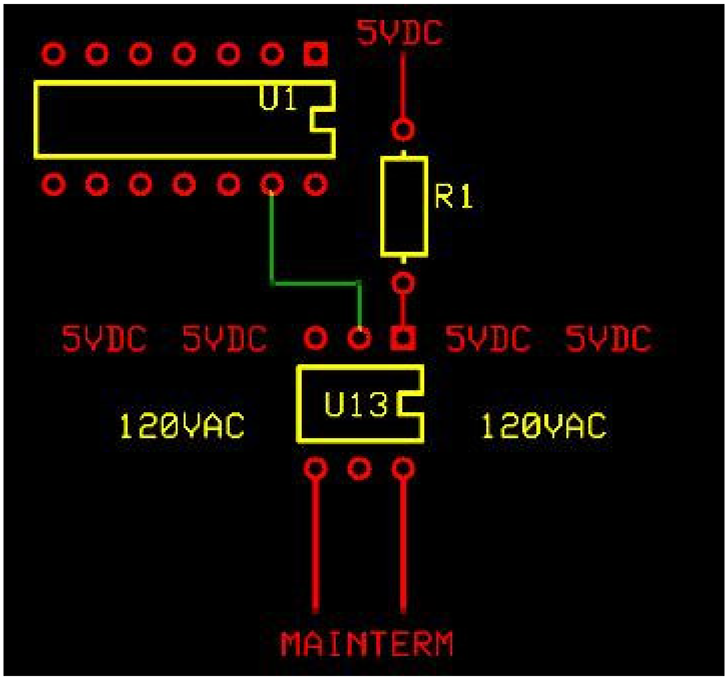
To summarize, the triac will only begin conducting when triggered and will continue conducting regardless of the state of the gate until the next zero voltage crossing point. The triac can be triggered in either the positive or negative part of the AC sine wave.

A helpful article titled "Thyristors and triacs - Ten golden rules for success in your application" can be found here:

http://www.web-ee.com/primers/files/AN_Golden_rules.pdf&e=7620

Rin Selection for the Triac. Sean Bowf (creator of the SSR circuit we are currently using) indicates he has had great results with 750 ohms and no known issues. I personally was not able to find a SIP (slim inline package which has many resistors in it) at exactly 750 ohms. Sean indicates that if you can not match exactly 750, go down in resistance rather than up. Here is the equation he recommends for selecting a value for R1. **5V applied power - 1.3 V input LED / .005 (opto IFT) = Resistor value** (or 740 ohms). Sean writes: "I think you might get away with 820 ohms...not sure. But you get someone that uses less then 5V (I saw some 4.5V wall warts), they might have a problem. I think you would be safer going smaller, as opposed to bigger."

Opto Isolator Triac Trigger



R1 = 780
U1 = PIC
U13 = MOC3023
The green trace is an example of a control signal from the PIC output PIN (which operates from 0 to 5VDC) that goes to the cathode of the Opto-Isolator and causes it to lower or raise the impedance between the two “Main Terminal” pins, causing the AC current to flow or not (lights go on and off! Now we are getting somewhere!)

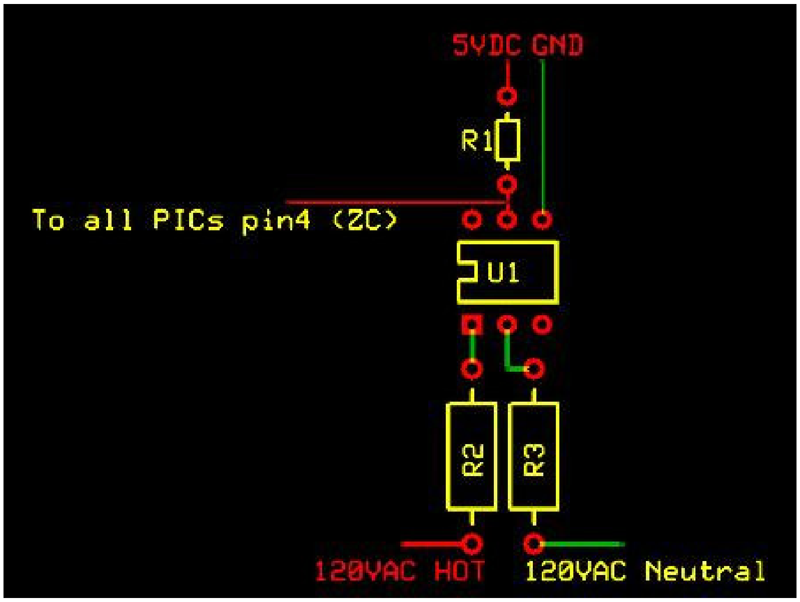
If you have done everything correctly, everything above the “5VDC” text will be low voltage, everything below the “120VAC” text will be high voltage.

(http://forums.parallax.com/forums/default.aspx?f=21&m=61194)

The triac trigger is a device that provides 5 V inputs to an internal LED (through a suitable current-limiting resistor) on one side and a 120 VAC optically triggered triac on the other side. This device electrically isolates the 120 VAC side of the circuit from the logic level side of the circuit. Using a triac to trigger the main triac solves the problem of triggering the main triac on the positive or negative voltage portion of the AC sine wave.

The opto-isolator in this design has a minimum activation current of 5mA. V=IR, thus you can see what a 680 ohm resistor was chosen based on a 5V system. This provides 7mA to the opto.

Zero Crossing detector (H11AA1)



R1 = 27K – quarter Watt
U1 = H11AA1
R2,R3 = 15K – halfWatt

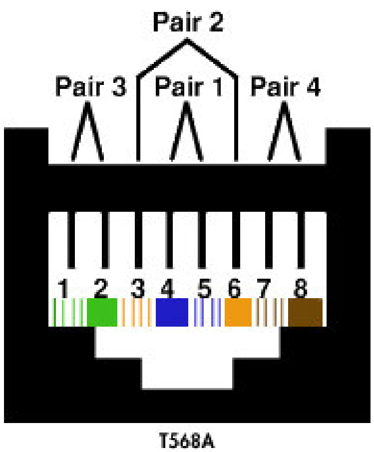
There are several approaches to detecting when the AC voltage is zero. It could be done by bringing 110VAC onto the board (and using the H11AA1 circuit as shown above, and used in the 16ch PicDimmed Renard with SSR design), by using a dedicated pin on the cat5 (That is, having a H11AA1 or another device located near the computer, and sending the ZC signal down the Cat5), or by using a CAT5 pin for the power input and zero-crossing at the same time.

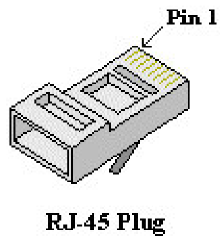
Renard doesn't have interrupt-on-change enabled, so there is no interrupt from the zero-crossing logic. Also, the input to the transistor circuit that has occasionally appeared in connection with Renard is a full-wave rectified AC signal, so the output of the transistor is very similar to the output of the H11AA1.
(http://forums.parallax.com/forums/default.aspx?f=21&m=61194)

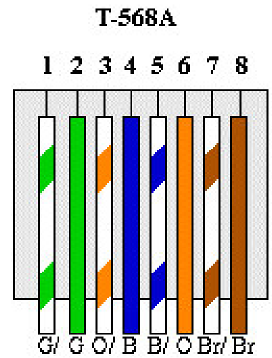
This device provides notification of when the AC voltage is zero. It consists of two cross-coupled LEDs on the AC side and a phototransistor on the logic level side. Thus the one or the other LEDs is on during any non-zero portion of the AC sine wave. Both LEDs are off at the zero voltage point. By adding a pull-up resistor to the photo transistor output on the logic level side of the device, the zero crossing detector will pull the output low during the non-zero portion of the sine wave and the pull-up resistor will pull the line up to 5 V when the photo transistor turns off. The zero crossing detector will produce a positive going pulse every 8.333 milliseconds (1/120 = .0083333). This device also electrically isolates the 120 VAC side of the circuit from the logic level side of the circuit.

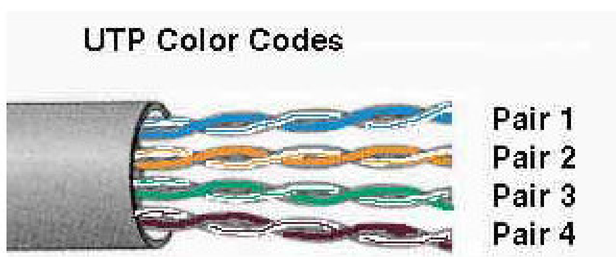
RJ45 Interface (CAT 5 specific – NOT CAT5e)

Clarification: This design will work just fine with Cat5e or Cat5. But the color scheme demonstrated below is for Cat5 NOT Cat5e. Just make sure you follow the NUMBERS if you are doing Cat5e. The colors are only correct for CAT5.









For this design:

Pair 1 = RS485 B is Blue(4), RS485 A is Blue-white(5)

Pair 2 = Orange(6) and Orange-White(3) are spares. (ZC in the future?) I have placed traces on the board such that a solder ball could be added to make these additional power and ground if needed. They probably will not be necessary given the low DC power consumption of the board. But you could solder them to make this board 100% RJ45 pin-out compatible with other co-op boards. In the easily modified case, Pin 3 becomes VCC, Pin 6 becomes ground. Even without this solder ball this board should still work in series with other Renard boards, it would just remove one of the power supply pins from the picture downstream from this board if you didn't make the change. In short: these could be used for anything you like. Look for via's labeled "3" and "6" near both RJ45 connectors.

Pair 3 = Green-white(1) and green(2) for ground (intuitive)

Pair 4 = Brown(8) and Brown-white(7) for >9 VDC < 15VDC (Input to 5V regulator). The reason the <15VDC rule is imposed is due to the use of the ESD protection MOV device listed in the parts list. If you use this MOV it has a maximum continuous operating voltage of 18VDC, thus I suggest never going over 15VDC continuously applied via the RS45 interface.

Capacitor Selection

For bypass caps on the PICS .1uF ceramic caps were chosen because their ESR is lower, and they are cheaper.

(Quoting Phil) For the caps around the regulator I would tend to use smaller values, simply because they are less expensive and can be lower ESR parts. The Fairchild datasheets for 7805 and 78L05 parts show 0.33 for the regulator input cap, and 0.1 uF for the output cap (for stability and transient response). For the ON-Semi LP2950 regulator both caps should be 1 uF, for which I would again use ceramic parts (although their price seems to be almost the same as for tantalum parts). The value of the input caps here assumes that the power input to the board is a DC value that never dips below about 7V (for 7805 or 78L05 parts) or below 5.5V (for the LP2950 part).

If someone is feeding the board with AC (or unfiltered DC) then the input cap for the regulator must be much higher, perhaps 100 uF or more (I haven't done the math yet for the high channel-count boards). Here is where I would put the tantalum caps in play (only use regular aluminum electrolytic's if the board stays at room temp, because they behave very poorly at freezing temps or below).

This board design assumes 5VDC is provided by an onboard regulator. 1uF ceramic capacitors were specified for use before and after the 5V regulator, for simplicity (lower part count), and to work in all cases where a 5V regulator is used.

Future Design Compliance

The following is a list of the minimum requirements desired in order to be compliant with the other boards that have been made, and therefore increase the value to all developers:

1. Output pin numbering of channels should be 1 to 8 linearly, no mixing up the numbers
2. It would be optimal if designs would take both RS232 and RS485 as input.
3. Debug features, LEDs, switches, etc
4. Work closely with Phil and others to insure your implementation of the pins on the RJ45 port will work with other designs.

Future Design Desires

1) Add status LEDs (for the PIC, not for each channel).
2) Lamp test switch.
3) Add external clock (for higher baud rates).
4) On-board programming capabilities. There are two possibilities here - add hardware from one of the el-cheapo designs so that you could start with a blank PIC, or use the code-Flash design which starts initially with a pre-programmed PIC).
5) Change to RS485 interface (requires some sort of controller to connect the PC to the dimmers).
6) DMX compatibility (full compatibility requires RS485 interface, different connectors, and some way explicit method of assigning an address to each PIC chip, and some non-trivial firmware changes). Again, this requires some sort of controller between the PC and the dimmers.

7) Bi-directional communications (the hardware part of this should be added when incorporating the RS485 interface), to enable adding some of the more advanced software features.

The Sean Bowf SSR

The part selection and layout of the SSR design was leveraged from Sean Bowf's design.
http://computerchristmas.com/index.phtml?link=how_to&HowToId=68

Coop Purchase Opportunity

Two prototype PCB's have been built and tested and are working perfectly at 25mS resolution and 38400 baud. (Supports up to 64 channels per serial port) I anticipate placing a order in mid February or early March. If you are interested in participating in a group purchase, drop me an email with your forecast for quantity.
jaunemaillot@gmail.com

Additional Resources

The main place to find support specifically for PIC based Christmas light controllers
http://picdimmer.17.forumer.com

The preferred software used with this setup
http://vixenlights.com/

Another good resource for how-to's and information related to Christmas light controllers
http://computerchristmas.com/

Revision History

1/28/2007 – Initial release
1/28/2007 – Added power hookup info
1/29/2007 – Added TPA and TPB in test point section
1/30/2007 - Added proto via's and lower layer surface mount ESD pads for both A/B 485 chips.
2/10/2007 – Lots of updates, final version. Design is complete and tested, document is compliant.
2/20/2007 – Added discussion on MPLAB and PICkit1 Flash starter kit software
2/21/2007 – Added "How to hook it all up"