# Simple Interface Rev 5

Tommy N. Tyler 26 November 2002

**1. REVISIONS** This version of the Simple interface is improved by using 1K resistors instead of 10K for R1 and R2. The instructions have been completely rewritten to make them clearer and easier to follow. The sections on Testing and Troubleshooting now incorporate error messages used in the current version of IR.exe.

**2. GENERAL DESCRIPTION** The Simple interface does not use any IC's or other active devices, but instead connects signal lines from the PC's parallel port directly to the remote control. It works with either 2-cell or 4-cell remotes, and is by far the easiest and least expensive interface to build. The downside is that in a small percentage of cases there may be problems in performance ranging from minor annoyances to complete failure. Generally, these problems are caused by an unfortunate combination of characteristics of the particular PC parallel port you are using and the model remote you are testing. In most cases there are "work arounds" that enable satisfactory operation, and these are discussed in the sections on Testing and Troubleshooting. Batteries should be installed in the remote to upload or download, even though the interface *may* pass the **Check Interface** test of IR.exe without them.

**3. PARTS REQUIRED** The following parts are all you need, except for the cable and 6-pin mating connector for JP1.

| ITEM | QTY | DESCRIPTION |
|---|---|---|
| 1. | 1 | DB25 male connector |
| 2. | 1 | DB25 connector hood, any size or shape |
| 3. | 2 | Resistor, 1K 1/4W (Radio Shack 271-1321) |
| 4. | 1 | Signal diode, 1N4148 or equivalent (Radio Shack 276-1122) |

The part of an interface that always has been, and still is, the most difficult to obtain is the 6-pin female connector that mates with JP1 in the remote. To keep these instructions short and focused on the rest of the project, they assume the reader is using a flat ribbon cable with suitable connector. Search the JP1 Group messages and files for references to connectors. In particular, you should download the file **6PinConnector.zip** from the **Files>Interface Designs** folder of the JP1 Group. It is strongly recommended that you make or buy this critical component as the first step before starting to build an interface.

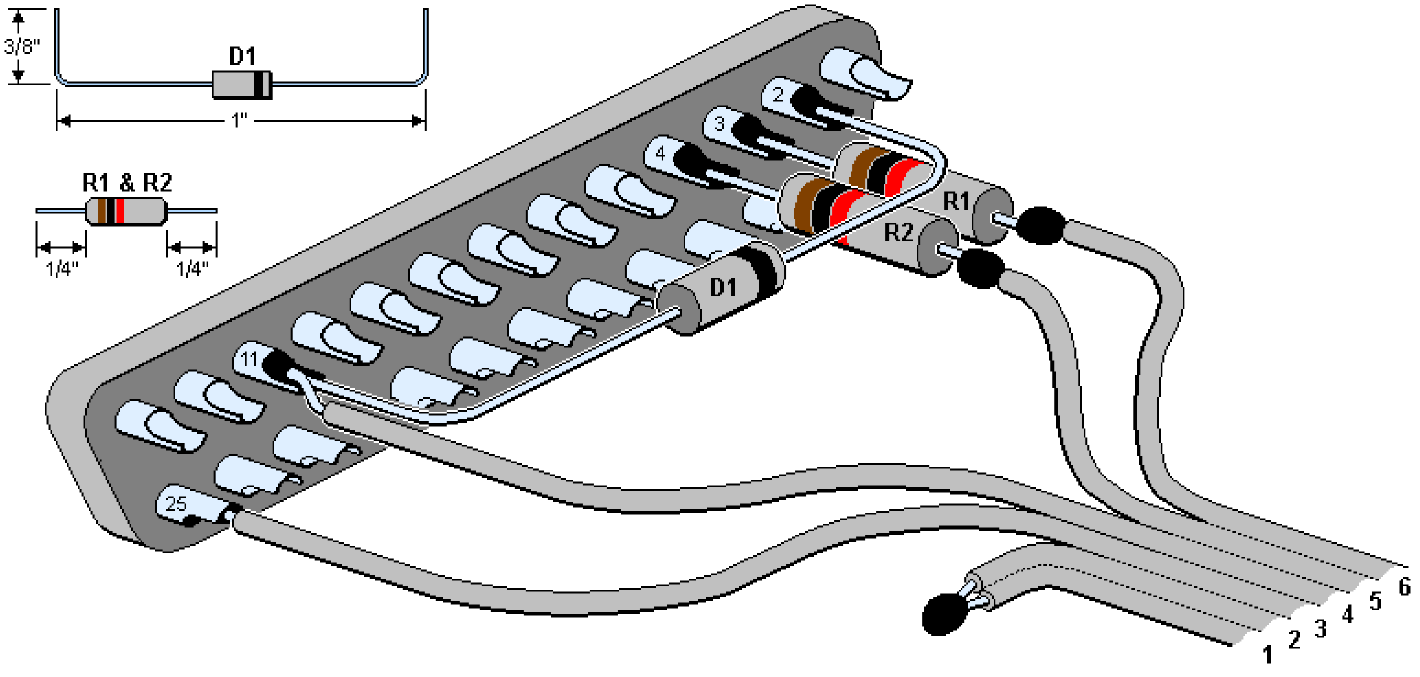

**Figure 1. Assembly of Cable and Components to DB25 Connector**

**4. ASSEMBLY** Figure 1 shows how the cable and components are soldered to the DB25 connector, along with some details of suggested ways to pre-cut and form the component leads so the diode lead rests on the bodies of the resistors and not their leads. The wire numbers refer to the pins of the JP1 connector they are attached to. Wires 1 and 2 of the cable are soldered to each other and nothing else. If you're using something other than a ribbon cable and IDC connector you have the option of connecting these pins together back at the JP1 end, and using a 4-conductor cable, as shown in Figure 3. The diode is the only polarized component in the assembly, and its banded end ***must*** be oriented toward pin 2 as shown.

In most computers pins 18 through 25 are all grounded inside the computer, and pin 3 of the JP1 connector must be connected to at least one of these grounded pins. Figure 1 shows the connection being made to pin 25, which was selected arbitrarily. That wire might just as well be soldered to pin 24, or 23, etc. But occasionally we run across a computer that does not have pin 25 grounded. How do you know, and what should you do about it?

You have at least three options. First, you can solder the wire to pin 25 as shown and not worry about it unless the interface doesn't work. The odds are probably 99 to 1 you won't have a problem. Second, you can measure the resistance from pin 25 to the chassis of your computer with an ohmmeter. It should be less than one ohm. If pin 25 doesn't appear to be grounded, find another pin in the group from 18 to 25 that is grounded, and use it. Third, just shotgun it by soldering a piece of wire to connect pins 18 through 25 as shown in Figure 2.

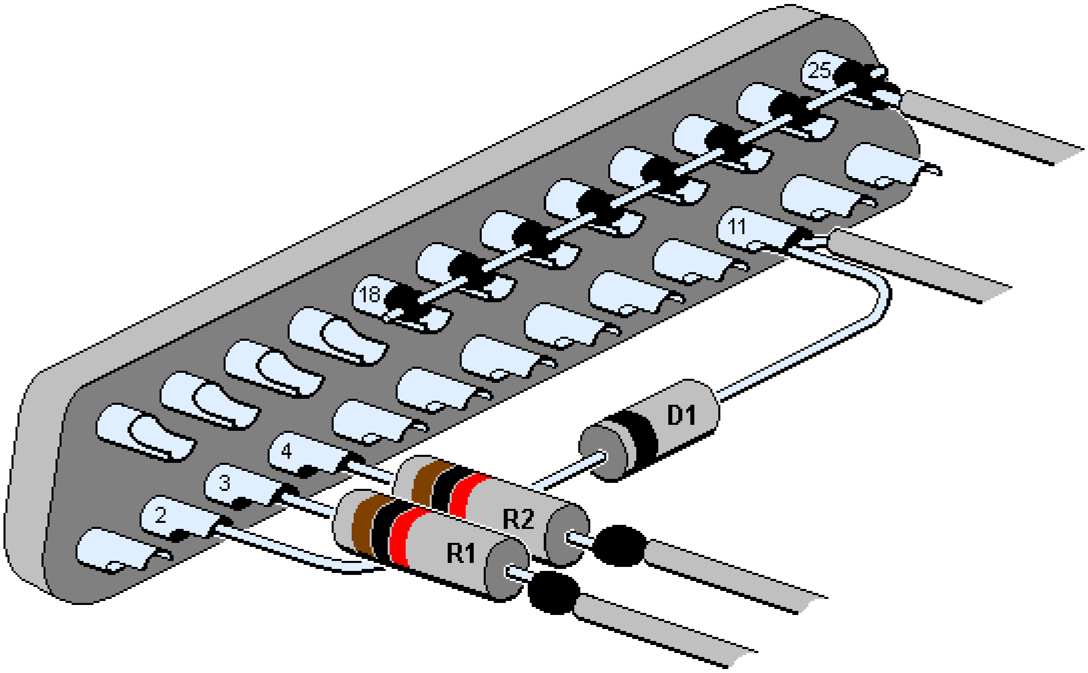

**Figure 2. View of Connector Showing Pins 18 Through 25 Connected**

When you have completed all soldering and double-checked everything for assembly errors, bad solder joints, shorts, etc., you can either go ahead and enclose the connector in its hood, or go to the section on Interface Testing and try it first, depending on your level of confidence. Figure 3 shows the end product. The picture shows a ribbon cable with IDC (Insulation Displacement Connector) that was made by cutting a larger one to a 10-pin size and using the middle six contacts, following the instructions in the download file **6PinConnector.zip** mentioned earlier. It also shows the option of using a 6-pin connector with solder pins and a 4-conductor cable. The type of cable you use is not critical, but don't make it more than six feet long.

Although not shown in Figure 3, you must provide some form of strain relief for the cable so the wires won't pull on their connections during handling. All hoods are supplied with a cable clamp of some sort. If you are using a jacketed cable that's smaller than the clamp, wrap it with a few layers of electrical tape so the clamp holds securely. When using a ribbon cable the best way to provide strain relief is to reverse one of the cable clamps so the two nest with each other. But don't over-tighten the clamp or it may squeeze through the ribbon cable insulation over time and short out the wires. Better yet, add some insulation material between the clamp and ribbon cable, such as a few layers of electrical tape.

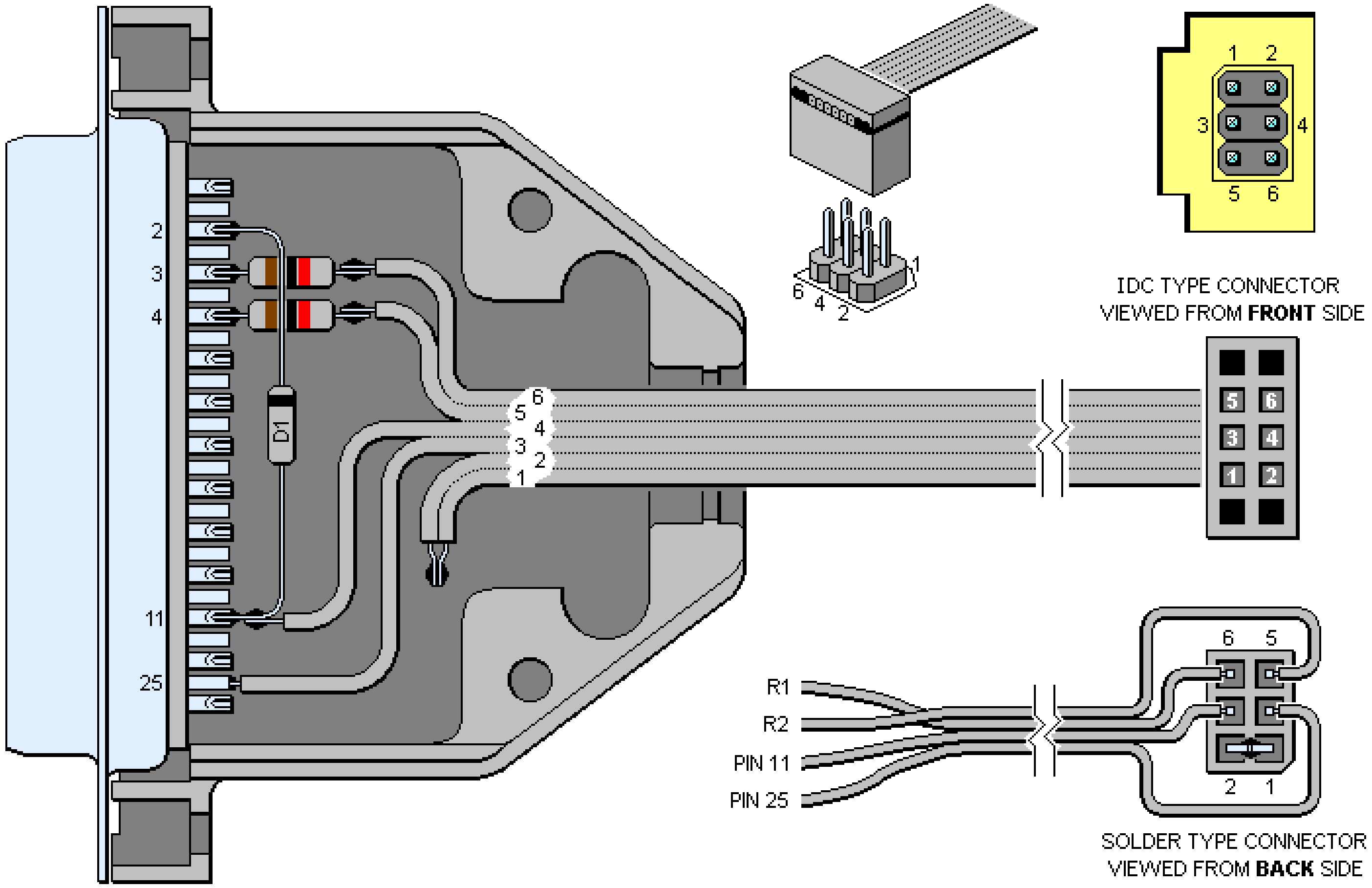

**Figure 3. Final Assembly of Interface**

Most assembly errors are made at the JP1 connector end of the cable because it's so easy to get confused and disoriented as to the numbering of the pins. Study Figure 3 carefully until you are sure your wires are connected to the proper positions of JP1. If you do any continuity testing with a meter DO NOT stick anything larger than a 1/4 watt resistor lead into the connector or you may risk opening up the contact so that it no longer makes a reliable connection with the pin in the remote.

**5. REMOTES WITHOUT 6-PIN CONNECTORS** If you are building this interface and plan to try and use it with a remote without opening up the remote and installing a 6-pin connector in the empty holes, don't be surprised if you have a lot of problems. It is true that some members have inserted a 6-pin header into their cable connector to convert it into a sort of male connector, and uploaded or downloaded while holding it against the remote's circuit board so that all six pins make contact simultaneously. But this is a risky proposition because those holes are slightly larger than the pins, with no resiliency to compensate for slight misalignment. To make contact with all six pins sometimes requires tedious trial-and-error twisting and tilting of the connector, which can damage the holes over time. Opening a remote isn't all that difficult if you follow the many instructions available, and if you are worried about damaging the circuit board maybe you can find a friend who is more experienced with a soldering iron.

**6. TESTING** This section outlines a systematic sequence of tests you should make before trying to use a newly built interface. Don't expect help from the Group unless you have followed all the suggestions here. Plug the interface into your computer and connect the cable to your remote. (Be sure there are batteries in it.) Start IR.exe, click on **Configuration > Transmission Delay**, and set the value to **1600**. Then click on **Configuration > Set Port** and set the port address to the correct value. If you are not sure what port address you're using you can just try the default address 378 and if that fails come back and try address 278. You won't damage anything by using the wrong address, but if there's another problem you may get an error message with both addresses. In that case you have no choice but to dig out your owner's manual and learn the correct address. If **Configuration > Auto Check** is not checked, activate it by clicking on it.

Click on **Configuration > Check Interface**, and **OK** to start the test. If you get a message that says **The remote interface check succeeded**, congratulations, you are done with testing. Go directly to the section entitled Making a Trial Download.

If you get an error message instead, don't panic. The tests IR.exe makes when you run the **Check Interface** test consist of two steps. First, IR performs a basic test whose function is just to verify that there is a live interface plugged into the PC at the specified parallel port. It does this by toggling the DØ output of the PC up and down a few times and making sure that signal loops through the interface and causes the S7 input of the PC to respond appropriately. Only if that test passes does IR proceed to a functional test to verify that there is actually a working remote connected to the interface. It does this by sending an address byte to the remote's EEPROM to see if it sends back an acknowledgement.

There are two error messages associated with the basic test ( **Always reading "1" from inverted input S7 . . .** and **Always reading "Ø" from inverted input S7 . . .** ) and either one can occur with or without the remote connected. There is one error message associated with the functional test ( **No response from remote** ). If you still get either of the first two messages even after disconnecting the cable from your remote, you may as well leave it disconnected until you solve that problem. When you no longer receive either of those two error messages you will receive a **No response** message, at which point you should reconnect the remote and continue testing.

There are a number of reasons you might receive any one of these error messages even when there's nothing wrong with the interface you've built. Probably 90 percent of all problems are external to the interface itself, another 9 percent are related to the wiring of the 6-pin connector or the way it's plugged in, and only 1 percent involve the construction under the hood. For example, if there's nothing connected to the parallel port address you've selected in config you'll get the **Always Ø** message. If you've selected a port address to which a printer or other peripheral is connected you'll get either an **Always Ø** or **Always 1** message, usually accompanied by a little note that says **Also, an unexpected value ($XX) was encountered in the status register**. If you plug the cable into the remote upside down you'll get the **Always 1** message. So the first thing to do before looking for errors in the interface is to double check your setup and configuration settings and check for possible errors in the position or orientation of the JP1 connector in the remote.

Except for the address, you don't have to worry about any other configuration parameters for your parallel port, such as IRQ, or whether it is configured as an SPP, EPP, or ECP port. IR.exe doesn't care about those things.

One good troubleshooting technique is to confirm positively that a given change causes a given effect by testing with and without the change several times. As an example, suppose you're getting a persistent **Always reading X** error, so you know you are not passing the basic test. Following the above suggestion, you decide to disconnect the remote to concentrate on this problem, and suddenly the failure mode changes to **No response**, which means you *have* passed the basic test! Good test procedure would be to try alternating with and without the remote connected several times, to make sure the failure pattern is consistent. This example would suggest that something occurs only when the remote is plugged in that is preventing the interface from passing the basic test. A possible cause might be that pins 3 and 4 of the JP1 connector are swapped. That would not interfere with passing the basic test with the remote disconnected, but pin 11 of the PC will be grounded any time the remote is connected, and the PC would interpret this as **Always reading "1"**. Although this example is not a very common occurrence, it illustrates the point that by repeating the test several times and observing that the **Always reading "1"** error only occurs when we plug into the remote, our attention is focused on that connector. Even if you lack the technical skill to deduce exactly what the problem is, you at least have an idea where to look. Confirmation like this also adds credibility to any request for assistance made to the Group.

**7. TROUBLESHOOTING** If you have followed the Interface Test instructions precisely but you're still having problems, this section contains more detailed technical information that should get you up and running. If the technical stuff makes your head spin just skim through it for possible ideas. I'll start with some information that helps explain the error messages a little more completely.

The DØ output (pin 2) is normally held high (near + 5 volts) by the PC, and momentarily driven low (near ground) from time to time when the PC is sending out data. High's are "1's" and low's are "Ø's". The S7 input (pin 11) is normally pulled up to + 5 volts by a resistor internal to the PC, but the input signal is inverted between the point where it enters the PC at pin 11 and the internal point where the PC actually reads it. That's why the error messages refer to "inverted input S7". Now let's add those two facts together. Normally the voltage on both pins 2 and 11 is high, and the PC is reading the inverted S7 value as low. When the PC sends out data it makes pin 2 go low and high in a sequential pattern. Each time pin 2 goes low it pulls pin 11 down toward ground through the diode, and when pin 2 returns high pin 11 returns high also. So when the output sequence at D0 is "1Ø1Ø" (high-low-high-low) the PC reads the inverted input at S7 as just the opposite, "Ø1Ø1" (low-high-low-high).

When the error message says **Always reading "Ø" from inverted input S7 even when putting "Ø" on output DØ** it means the PC thinks input pin 11 is stuck high because it can't make it go low by pulling it down with output pin 2. The reason might be that something is causing + 5 volts to be applied directly to pin 11, or it might be that the circuit from pin 2 through the diode to pin 11 is not working properly.

When the error message says **Always reading "1" from inverted input S7 even when putting "1" on output DØ** it means the PC thinks pin 11 is stuck low because when the PC makes output pin 2 high something else continues to hold pin 11 down. The reason might be that there is a short to ground somewhere in the wiring from pin 11 to pin 4 of the JP1 connector.

Hopefully, from the above discussion you can draw a few conclusions about the interface diode that connects pins 2 and 11. If the diode is open circuited or installed backwards the DØoutput will be unable to pull down the S7 input, and you'll get an **Always "Ø"** error. If the diode is shorted (the way most diodes fail) you'll still be able to pass the basic test, but you'll get a **No response** failure when the remote tries to reply to its address. However, if the error message says **Always "1"** the diode can't be involved in the problem because there's no way it can be the cause of holding the S7 input low while the PC is keeping the DØ output high, even if it is shorted, reversed, or open.

Fortunately, the interface has so few parts that finding an assembly error by visual inspection is usually not very difficult. Tests requiring access to the internals of the interface while operating are easier to make if you use an extender cable to bring the interface out where you can get to it. Just be sure *not* to use a null modem cable as these have pins 2 and 3 swapped from one end to the other, and many don't even have connections to pins 11 and 25. If you use an extender cable make absolutely sure it has continuity between pins 2, 3, 4, 11, and 25 from one end to the other. None of the other pins matter.

Your first task is to get to a point where you can pass the basic test with the remote disconnected, i.e., you no longer get an **Always reading "X"** error message. It is important for you to understand that the only component in the interface involved in passing the basic test is the diode (D1), because it's the only component in the signal path the PC uses to make the basic test. It is not unheard of for mis-marked diodes to be sold at bargain prices to disreputable suppliers. For a persistent **Always reading "Ø"** error there is a simple way to determine if the diode is faulty, which is to run the **Check Interface** test while shorting out the diode with a piece of wire or a paper clip. If the basic test passes when the diode is shorted but fails when it is not, replace the diode. If the test fails even when the diode is shorted the possibilities are narrowed considerably. Perhaps what you think are pins 2 and 11 of the DB25 connector are not really pins 2 and 11, or they are not making contact with the PC connector.

As a last resort unplug the interface from the PC, jumper pins 2 and 11 of the parallel port connector with a bent paper clip as shown in Figure 4, and run the **Check Interface** test. If you still get an **Always reading "X"** message instead of a **No response** message there's a problem with your computer or the way your parallel port is setup.

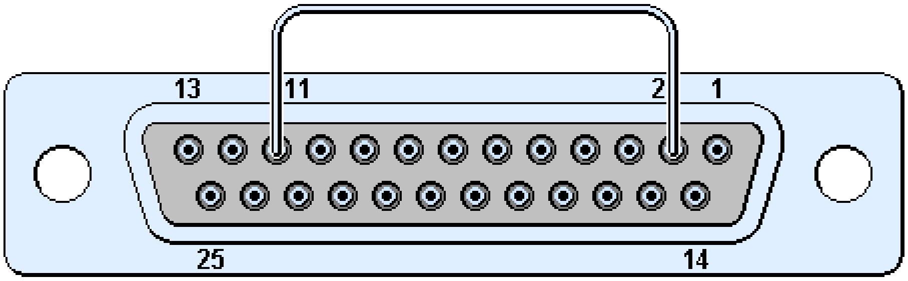

**Figure 4. Jumper for Loop-Back Test of Parallel Port Connector**

When you get a **No response** error message it means that the IR program can talk to the interface but is unable to communicate with the remote. The odds are still 10-to-1 that the problem is with your cable and/or JP1 connector. So first make sure the connections from your ribbon cable (or whatever cable wires you are using) to the female pins in the JP1 connector are intact. You should check continuity between the end of each wire inside the interface hood and its corresponding contact inside the JP1 connector. Always use the lead wire of a 1/4-watt resistor as a probe for inserting into a JP1 contact, nothing larger.

The odds are extremely small that the problem is in the remote itself, if it is otherwise working properly. That is, unless you've installed the 6-pin connector yourself. In that case you might double check your work for shorts or poor solder joints. A common mistake is to use too much heat when installing the connector, causing the plastic to soften so that the pins are crooked and make poor contact. It's best to use a mating connector to hold the pins straight while soldering.

If you used ribbon cable and an IDC connector in your construction there are several traps you may have fallen into. Since most ribbon cable wires aren't color-coded it's easy to get them mixed up or to swap wire 1 for wire 6. And if the cable isn't positioned carefully before squeezing the cover down, some of the tiny contact barbs can get bent over or miss the wires completely. One of the subtlest mistakes can occur if you make your IDC connector by sawing off a piece of a longer one. Figure 5 shows the standard arrangement of wires and pins, using a 14-pin IDC connector as an example. The left portion of the illustration shows an overhead view of the header with its two rows of 1/40" square pins spaced 1/10" in both directions, the mating connector with top removed so you can see the barbs that pierce the wire insulation, and the ribbon cable. All IDC connectors have an even number of pins, with the barbs staggered as shown because the wires are spaced every 1/20" apart. The actual parts are not marked with pin numbers, and the connector can be reversed end for end without changing anything. The pin numbers in Figure 5 show the most common convention, which is that when holding the connector over the header with the ribbon cable exiting toward you, pin 1 of the header is the near left corner one, and the wire on the left is number 1. That's the way the pins of the JP1 header are arranged. Some IDC connectors have a strain relief that loops the cable back over the top of the connector and clamps it there with a piece of plastic. This obviously reverses the direction rule. Find pin 1 of your connector and mark that corner with a drop of fingernail polish.

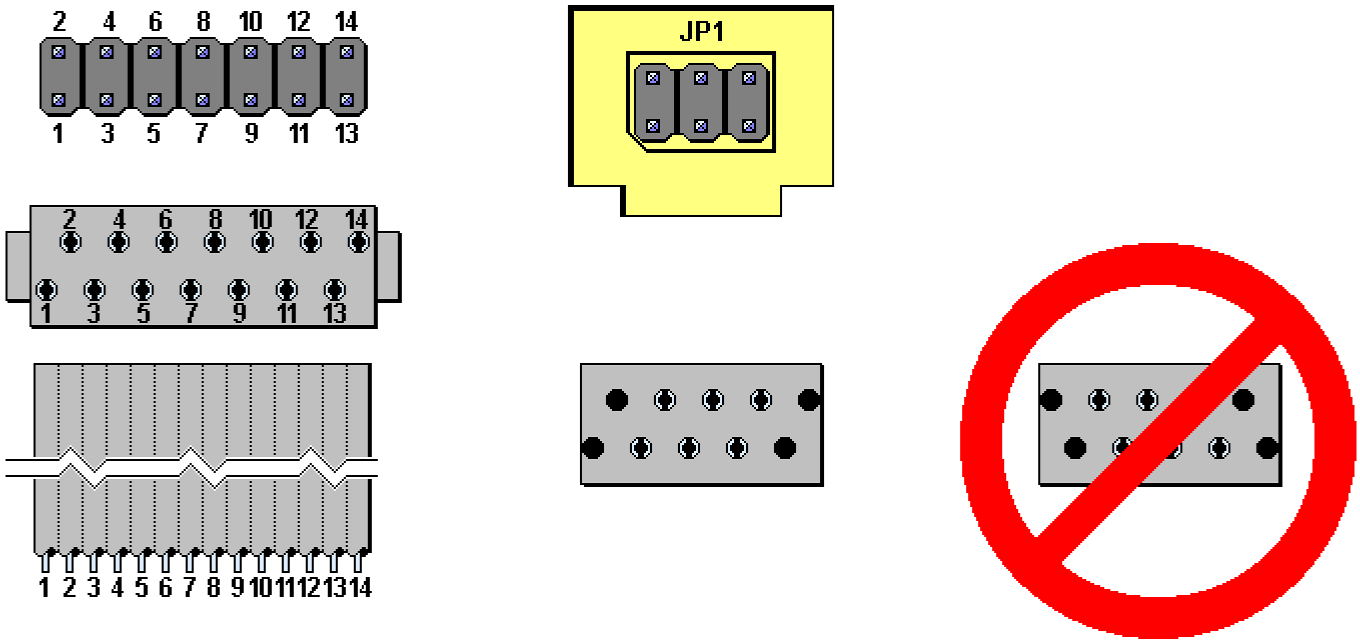

**Figure 5. Standard Numbering of IDC Connector Pins**

When you cut a longer connector to make a shorter one, in order for wire number 1 to make contact with pin number 1 you should always try to leave one end uncut and cut the other end so as to have an even number of sockets. You'll actually cut through one of the wire piercing barbs when you do this, which is why it's recommended you use a 10-pin housing with the outer barbs removed for a 6-pin connector. This also makes the connector wide enough that you can't inadvertently plug it in off center. If you somehow wind up cutting both ends be careful. Figure 5 shows a 6-pin connector made wrongly by cutting a very long connector between pins 11 and 12, and between pins 21 and 22. It "starts on the wrong foot" so to speak, and there's no way it can be made to work. And you can't fix the mistake by turning the connector around and re-assembling it to the cable. That connector is scrap.

**8. MAKING A TRIAL DOWNLOAD** Once you can pass the **Check Interface** test you're ready for a trial download. Even if you can't pass the test you won't hurt anything if you try a download, but you will undoubtedly just get the same error message since IR will make the same tests if the **Auto Check** feature is check marked. The Auto Check option causes IR to make the interface tests automatically at the start of every upload or download, as a means of continually monitoring the proper functioning of the interface. It is activated or deactivated by clicking on **Configuration > Auto Check**, and I recommend that you leave it activated except when experimenting to see if you can download without passing the interface test.

It doesn't really make any difference which page of IR is displayed while you are downloading, and they are all blank anyway. Click on **Remote > Download from Remote**. That should bring up a very important confirmation message that says **Ready to download from remote**.

**CAUTION:** If you see a longer message that says **Ready to upload to remote. This will completely overwrite the programming currently in the remote**, you have clicked on **Upload** by mistake, and you should click on the **Cancel** button immediately. Never, ever upload any data to any remote until you have first downloaded and saved its present data so that you can restore the original programming if necessary.

Click on **OK** to start the download. If all goes well you will see the default name **Remote Control Programmer** in the title bar replaced with the name of your remote (e.g., **URC7200 Cinema 7**) and in a second or two the small red dot in the lower left corner of the window will blink twice to signal the completion of download.

If you get a message that says **No definition file exists for this remote. . . .** it usually means there is no RDF (Remote Definition File) for your model remote in the directory with IR.exe. Don't worry about that right now. When you click on **Yes** you should still receive a download of the uninterpreted raw data. You just won't be able to do any programming until IR can identify your remote. When you download IR.exe it comes bundled with RDF files for all the model remotes IR supports. So the first thing to try is downloading the latest revision of IR.exe. If the RDF file is not in there you'll have to ask for help from the Group

Click on the **Raw Data** tab to view the EEPROM contents. If the data is all "**FF**"s or all "**ØØ**"s the problem may be something other than a missing RDF. Sometimes the only way to rectify this problem is to do a full manual reset of the remote (**Code 981**) and try another download. Unfortunately, the reset will cause all programming in the remote to be erased.

Sometimes an interface will pass the **Check Interface** test completely but give a **No response** error message when you try to download. The most likely cause is incompatibility between your PC and remote. Unfortunately, the incompatibility problem is partially dependant on the characteristics of the specific EEPROM in the remote, which is one reason why another person building the same interface for the same remote might get different results than you do. If you have another remote you can try, it may behave differently.

Make sure you are not running the interface any faster than a **Transmission Delay** setting of **1600**. To make sure the response time of your computer and cable is not a problem; try setting the **Transmission Delay** to a ridiculous value, such as **16000**. That's running in super slow motion (download of a 1K EEPROM will take about 12 seconds) so unless you get a miraculous cure set it back to **1600**. In case you are interested, the speed reduction factor is equal to the **Transmission Delay** setting divided by 100, plus 1. So a setting of **1600** is running the program 17 times slower than it runs at its maximum speed with a setting of zero.

The most common incompatibility problem is that the PC is operating at one voltage and the remote at another. For example, if a 2-cell remote with old batteries is operating at 2 volts, the PC's DØ output might not be able to pull the SDA line quite low enough through diode D1 to qualify as a "Ø". This situation can be improved by using a Shottky type diode such as the BAT83 or 1N5818, which has a forward voltage drop a few tenths less than an ordinary diode. Those are inexpensive components, but they're not available from Radio Shack, and must be obtained from an electronic parts supplier such as Mouser, Digi-Key, or Jameco Electronics. You might also be able to reduce the voltage mis-match just enough to cure the problem by installing fresh batteries in the remote.

The opposite situation occurs when trying to download a 4-cell remote operating at about 6 volts with a laptop PC that has an output voltage of 3.5 volts or less, because the PC's D1 output might not be able to pull the SCL line quite high enough through resistor R1 to qualify as a "1". Here the problem is made worse with fresh batteries. A possible solution is to use worn-out batteries, or operate the remote on only three of its four cells, as explained in the section on Permanent Reset.

There are many error messages that can occur during a download that relate to the content of the download (Header read failure, Checksum error, etc.) but those are beyond the scope of these instructions. This project concludes when you have a working interface.

**9. PERMANENT RESET** Sometimes a remote will remain permanently reset as long as it is connected to the interface. The symptoms of this are: (1) You can't operate the remote while it is connected. (2) The remote's red LED doesn't blink twice following **Check Interface**, **Upload**, or **Download**. (3) The red LED blinks twice whenever the remote is disconnected. There are two causes for this problem. One is the voltage mis-match situation discussed previously. If you are working with a 6-volt remote with fresh batteries, and your PC's output voltage is not high enough to release the reset signal on the remote's processor, the processor will remain reset as long as the remote is connected. The important thing to remember is that this shortcoming does not prevent you from using the interface to upload or download because the processor must be held reset during those operations anyway. It simply means the JP1 connector must be unplugged before the remote can be operated, something most people do routinely.

The reduction of resistor R2 in this Rev 5 version of the interface should greatly reduce the incidence of permanent reset, but there is not enough evidence yet to confirm this, particularly when used with laptop PC's. If you absolutely must be able to operate the remote while it is still connected to the interface you can try reducing the operating voltage of a 4-cell remote to 4.5 volts by replacing one cell with a dead battery or a dummy. This will result in some reduction in maximum operating distance, so it may not be a totally satisfactory solution. You can make a dummy AAA size battery by cutting a 1-3/4 inch length of 3/8 inch diameter wooden dowel rod and wrapping it with aluminum foil or aluminum (not cloth) duct tape.

**10. SCHEMATIC AND THEORY OF OPERATION** As shown in Figure 6, the RESET line is controlled directly by the D2 output of the parallel port through resistor R2, which limits current into the reset pin in case the remote is operating at a lower voltage than the computer. The SCL line is controlled by the D1 output through resistor R1. The PC can send clock signals through R1 to the EEPROM while the remote is reset, and the remote can send clock signals to the EEPROM while the PC is holding D1 high and R1 acts as a pull-up. The SDA line is controlled by the DØ output through the diode. When DØ is high the diode is reverse biased. Under this condition the processor can send and receive data to the EEPROM on the SDA line, and the EEPROM can send data to the PC on the S7 line. When the processor output is in tri-state, a low on DØ will pull down on the SDA line through the diode, and a high will let R19 pull up on it. That's the way the PC sends data to the EEPROM.

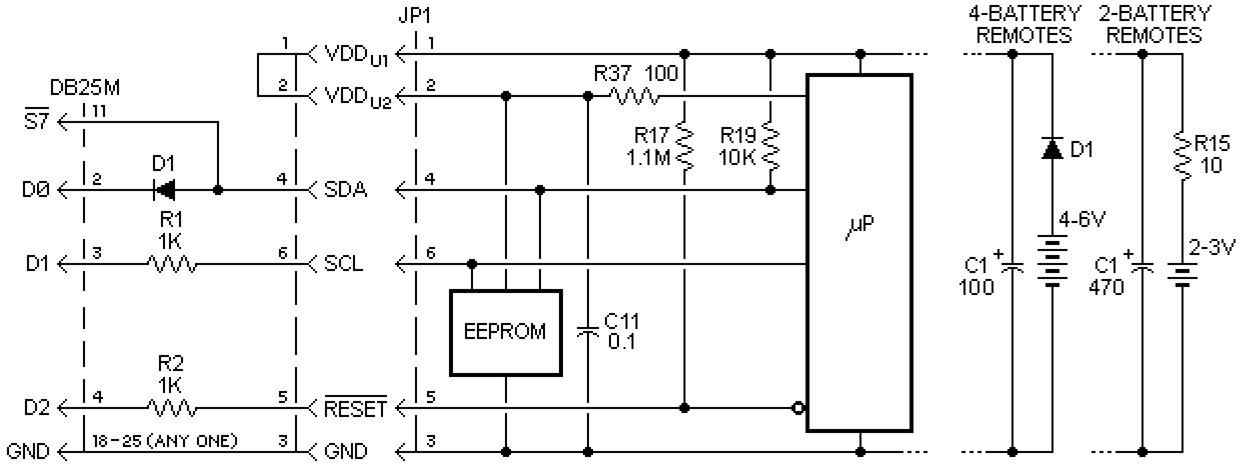

**Figure 6. Schematic of Simple Interface Connected to Remote**